

Technology Diffusion in Communication Networks

Sharon Goldberg
Boston University
goldbe@cs.bu.edu *

Zhenming Liu
Harvard University
zliu@eecs.harvard.edu †

February 28, 2012

Abstract

In the rich and growing literature on diffusion and cascade effects in social networks, it is assumed that a node's actions are influenced only by its *immediate neighbors* in the social network. However, there are other contexts in which this highly-local view of influence is not applicable. The diffusion of technologies in communication networks is one important example; here, a node's actions should also be influenced by remote nodes that it can communicate with using the new technology.

We propose a new model of technology diffusion inspired by the networking literature on this topic. Given the communication network $G(V, E)$, we assume that node u *activates* (*i.e.*, deploys the new technology) when it is adjacent to a *connected component* of active nodes in G of size exceeding node u 's *threshold* $\theta(u)$. We focus on an algorithmic problem that is well understood in the context of social networks, but thus far has only heuristic solutions in the context of communication networks: determining the *smallest seedset* of early adopter nodes, that once activated, cause a cascade that eventually causes all other nodes in the network to activate as well. Our main result is a near-optimal approximation algorithm that returns a seedset that is an $O(r\ell \log |V|)$ -factor larger than the optimal seedset, where r is the graph diameter and each node's threshold can take on one of at most ℓ possible values. Our results highlight the substantial algorithmic difference between our problem and the work in diffusion on social networks.

Contents

1	Introduction	3
1.1	Our setting.	4
1.2	Our results.	4
1.3	Relationship to the linear threshold model in social networks.	5
1.4	Related networking research	6
2	Formal Statement of Our Model	6

*Supported by NSF grant S-1017907.

†Supported by NSF grant CCF-0915922.

3	Approximation Algorithm	7
3.1	Highlights of our algorithm	7
3.1.1	Linearization & formulating the IP	7
3.1.2	Network flows & randomized rounding	8
3.2	Integer program	10
3.3	Correctness of the IP	11
3.4	Relaxing and rounding algorithm	12
3.4.1	Properties of the rounding procedure.	12
3.4.2	Overview of the the rounding procedure	12
3.4.3	Reconstructing the preliminary seedset S_0 and activation function T_0	13
3.4.4	Reconciliation procedure	14
3.5	Correctness of the approximation algorithm	15
4	Lower bounds	17
5	Going beyond heuristics.	18
A	The simple integer program	22
B	Some Examples	23
C	Optimal connected activation sequences provide a 2-approximation	27
D	Reduction to Set Cover	29
E	Connectivity implies dependence on graph diameter r	31
F	Our problem is neither submodular nor supermodular	32
F.1	Fixed threshold case	32
F.1.1	The influence function is not submodular.	33
F.1.2	The influence function is not supermodular.	33
F.2	Randomized threshold case	34
F.2.1	The influence function is not submodular.	35
F.2.2	The influence function is not supermodular.	36
G	Probability review	36
H	Lemmas to prove Theorem 3.3	37
H.1	Size of preliminary seedset S_0	37
H.2	Feasibility of preliminary activation sequence T_0	37
H.3	Connectivity Lemma for the activation sequence.	40
H.4	Partial consistency.	41
H.5	Proof of Gap Size Lemma 3.8	41
H.6	Proof of Lemma 3.9	42
H.6.1	Omitted proofs from Proof of Lemma H.7	46
H.6.2	Obtaining Lemma 3.9 from Lemma H.7	47

1 Introduction

Cascade effects provide a simple and effective way to drive global diffusion of a new technology in a network: after a few well chosen *seed* nodes are convinced to adopt the technology, more and more nodes make local decisions to adopt the technology until eventually everyone in the network has adopted it. Given the complexity and expense involved in persuading a large, dispersed network of nodes to adopt a new technology, a particularly important algorithmic problem is to determine the *smallest* possible seedset of early adopter nodes, and thus also the “cheapest” way to drive a cascade that leads to global adoption [11, 23].

Diffusion models are predicated on a model of node *utility*; namely, the benefit an individual node obtains when it decides to adopt the technology. In the rich literature on cascade effects in social networks (see *e.g.*, [15, 30, 28, 11, 23] and subsequent works), the model of node utility is highly local — it depends on only a node’s “friends” or immediate neighbors in the social network. However, there are many interesting contexts where this highly-local model of utility is not applicable.

Communication networks. We are particularly inspired by the example of diffusion of communications technologies in networks like the Internet; here, a node’s utility should depend not only its immediate neighbors, but also with on the number of (possibly distant) nodes that it can communicate with using the new technology. There has been significant interest in the networking community in the impact of cascade effects on technology diffusion (see Section 1.4). Motivated by this research, in this paper we propose a new model of technology diffusion. Central to our model is the idea of *non-local* utility, which is present in much of the literature on communication technologies *e.g.*, [6, 3, 14, 29, 20, 16, 21].

Utility as connected components. We say a node is *inactive* if it uses a older version of the technology, and *activates* once it deploys its new, improved version. In our model, node utility depends on the size of *connected components of active nodes* adjacent to node u in G , *i.e.*, on the size of the connected component containing u in the subgraph of $G(V, E)$ induced by $\{u\} \cup \{v : v \in V, \text{Node } v \text{ is active}\}$. This model captures the following two natural ideas:

1. A pair of nodes u, v may communicate using the new technology only if there is path from u to v in G consisting only of active nodes. This property characterizes many important networking technologies (see Section 1.4).
2. A node’s utility should depend on the number of other nodes it can communicate with using the new technology. This idea is known in the popular literature as Metcalfe’s Law, which states that utility that a single user gets from being part of a network of n users scales as n [27], and in also line with traditional ideas in economics, *e.g.*, [22]: “[t]he benefits that a consumer derives ... depend on how many other consumers ultimately purchase compatible units, ... in other words, ... [it] depends only on the final network sizes.”

Given this model of node utility, we consider the algorithmic problem of choosing the smallest seedset of early adopter nodes, that once activated, can cause a cascade that leads to global adoption of the new technology. This problem is particularly important in the context communication networks, where nodes typically represent profit-maximizing Internet service providers that must be convinced by governments or standards bodies to adopt a new technology (see *e.g.*, [6, 3, 14, 29]).

1.1 Our setting.

We assume that the underlying network structure $G(V, E)$ is fixed, and consider a progressive technology diffusion process: a node starts out as inactive (using a older version of the technology) and activates (adopts its new, improved version) once it obtains sufficient utility from the new technology. Once a node is active, it can never become inactive. To model the cost of technology deployment, we associate a threshold $\theta(u)$ with each node u that determines how large its utility should be before it is willing to activate. We assume that a node u activates when its utility exceeds $\theta(u)$, *i.e.*, if the connected component containing node u in the subgraph induced by $\{v : v \in V, \text{Node } v \text{ is active}\} \cup \{u\}$ has size at least $\theta(u)$.¹

Optimization problem. Given graph $G = \{V, E\}$ and a deterministic threshold function $\theta : V \rightarrow \{0, \dots, |V|\}$, our goal is to find a seedset S that is *feasible*, *i.e.*, when the nodes in S are activated *every other node* in the graph eventually activates as well.

1.2 Our results.

Our main result, stated precisely as Theorem 3.3, is an approximation algorithm:

Theorem 1.1 (Main result). *Consider a technology diffusion problem $\{G(V, E), \theta\}$ where the optimal seed set has size opt , the graph has diameter r (*i.e.*, r is the length of “longest shortest path” in G), and there are at most ℓ possible threshold values, *i.e.*, $\theta : V \rightarrow \{\theta_1, \dots, \theta_\ell\}$. Then there exists a polynomial running time algorithm that returns a feasible seedset S of size $O(r\ell \log |V| \text{opt})$.*

Exposition. Our algorithm, which uses linear programming, is based on two key ideas discussed in detail in Section 3.1:

1. *Linearization.* The non-local nature of our utility function makes it challenging to encode our problem as an integer program. Nevertheless, we observe that this function can be encoded using only local constraints if we restrict our search space to seedsets that give rise to *connected activation sequences*, *i.e.*, seedsets that ensure that set of active node induce a connected subgraph of G at every point in the diffusion process. We then show that this restriction means that our IP must return a 2-approximation to the optimal seedset.
2. *Randomized rounding using network flows.* Given the relaxation of the IP, we circumvent a potentially large integrality gap (Appendix A) by designing a novel randomized rounding algorithm that simultaneously interprets the fractional values returned by the linear program (LP) as *network flows* and probabilities. For this to work, we need to further restrict our search space; now we require *the seedset itself to be connected*, *i.e.*, the nodes in the seedset induce a connected subgraph of G , and thus can obtain an $O(r\ell \log |V|)$ approximation to the optimal seedset.

On the optimality of our results. For the wide range of problem instances where $r, \ell = O(\log |V|)$, our algorithm presents a $\tilde{O}(1)$ -approximation. The following lower bound (Section 4) indicates that our results are also near optimal for these instances:

¹Note that we can accommodate models of network value other than Metcalfe’s Law [27] by scaling the thresholds, *e.g.*, for the Odlyzko-Tilly Law [5] replace θ with e^θ .

Lemma 1.2. *The technology diffusion optimization problem does not admit any $o(\ln |V|)$ -approximation algorithm, even when the graph has a constant diameter r , and the number of threshold values is a constant ℓ . Furthermore, the result holds even if we require the seedset to be connected.*

Indeed, r and ℓ are $O(\log |V|)$ in many interesting settings:

Graph diameter r . Empirical graphs of communication networks like the Internet exhibit very small diameter: the autonomous system graph used in [14] has 37K nodes and diameter 11, while a router-level (skitter) graph from 2005 has 1.7M nodes and diameter 25 [26]. In fact, [26] has provided empirical evidence that these graph diameters actually *shrink* as the graph grows. Moreover, there is a large class of random graphs that have diameter $r = O(\log |V|)$, *e.g.*, Erdos-Renyi random graph family, the preferential attachment graph family [1], *etc.*

We can also show that our algorithm’s dependence on r follows because it is restricted to returning connected seedsets. The following unconditional lower bound suggests that circumventing polynomial dependencies on r requires an algorithm that can return disconnected seedsets, and likely also a different set of techniques:

Lemma 1.3. *For any fixed integer r , there exists an instance of technology diffusion problem $\{G, \theta\}$ such that (a) the diameter of G is $\Theta(r)$, and (b) the optimal connected seedset is at least $\Omega(r)$ larger than the optimal seedset.*

Threshold granularity ℓ . Parameter ℓ is a natural restriction on the granularity of the threshold function θ . [14] have argued that it is natural to restrict the granularity of θ , given the difficulty of obtaining empirical data on technology deployment costs relative to utility, for every single node in the graph. Indeed, the literature often deals with this difficulty by simply assuming that all nodes have the same θ values [28, 29, 14, 6], or by drawing θ randomly from some distribution [23].

Beyond heuristics. Given the prevalence of heuristics in the literature (see Section 1.4), one might wonder if the seedsets returned by our algorithms are actually any better than heuristic solutions. To give evidence that our approach finds solutions that are not found by heuristics, in Section 5 we run our IP on a few small problem instances and find that it does indeed return solution that are different (and often substantially better) than several natural heuristics.

1.3 Relationship to the linear threshold model in social networks.

One inspiration for our model was the *linear threshold model* for social networks, articulated in [23] and appearing in many other works. Indeed, we diverge from the linear threshold model only in our choice of utility function; our’s is non-local, while theirs assumes a node’s utility is given by the (weighted) sum of its active *neighbors* in G . Despite the superficial similarities, there is a substantial algorithmic difference between these models. [7] considers finding an optimal feasible seedset (*i.e.*, the smallest seedset that activates all nodes in graph) in the linear threshold model in social networks where thresholds are fixed, and deterministic, and shows that it is NP-hard to find a seedset of size $O(2^{\log^{1-\epsilon} |V|}) \cdot \text{opt}$ for any $\epsilon > 0$; his result holds even if $r, \ell = O(1)$. In contrast, if $r, \ell = \tilde{O}(1)$, then our main result shows that our model admits a $\tilde{O}(1)$ -approximation algorithm.

Moreover, [23] worked around these discouraging approximation-hardness result by assuming that thresholds were chosen uniformly and independently at random *after* the seedset was selected. This way, they could show the submodularity of the “influence function”, *i.e.*, the expected number of nodes activated by a seedset S , and therefore use greedy algorithms to find a seedset of size

$(1 - 1/e - \epsilon) \cdot \text{opt}$. In contrast, we show in Appendix F that this submodularity property fails to hold in our setting, highlighting another difference between our model and theirs. In fact, we show our influence function is neither submodular nor supermodular, even if we (a) randomize the thresholds as in [23], or limit ourselves to (b) graphs of constant radius, or (c) a constant number of threshold values. Moreover, we see neither diminishing, nor increasing marginal returns even if we restrict ourselves to (d) connected seedsets.

1.4 Related networking research

In addition to the long line of work on diffusion in social networks (*e.g.*, [15, 30, 28, 11, 23] and many others), the networking community has been grappling with the problem of technology upgrades in the Internet for many years, see *e.g.*, [9, 6, 3, 14, 29, 20, 13, 12, 16, 21]. A large number networking technologies are characterized by the property we described above: that an a pair of active nodes can only communicate using the new technology if they have a path between them consisting only of active nodes. The most obvious example is secure Internet routing [24, 25]. Here, cryptographically-signed routing messages may only be propagated on paths where each and every node is secure [25, 14]. This property is also shared by protocols like interdomain Quality of Service (QoS) [18], fault localization [4], denial of service (DoS) prevention [32], and, to a lesser extent, IPv6 [10]².

A number of works have used simulation studies to understand the relationship between seedset selection and cascading technology adoption *e.g.*, [29, 6, 14, 21]. Most of these works, with the exception of [14], have sidestepped the question of choosing an optimal seedset and have gone directly to using heuristics (often, “choose the high-degree nodes”). [14] study secure routing protocol deployment in realistic routing model, and after showing that it is NP-hard to find a constant approximation of an optimal seedset, move on to heuristics as well. While our work presents a more stylized model of the diffusion process, to our knowledge, it is also the first approximation algorithm to provide worst-case guarantees when utilities are non-local.

2 Formal Statement of Our Model

Definition 2.1 (Technology diffusion process). *Let $G = \{V, E\}$ be a connected undirected graph. Let S be the seedset, an arbitrary subset of V . Let θ be the threshold function of G , which maps V to $\{\theta_1, \dots, \theta_\ell\}$ with θ_i are all in $\{2, \dots, n\}$. The technology diffusion process on $\{G, \theta\}$ with respect to the seedset S is a family of functions $\{f_t : V \rightarrow \{0, 1\} \mid t \in \mathbf{N}\}$ such that*

- When $t = 1$, $f_1(v) = \begin{cases} 1 & \text{if } v \in S \\ 0 & \text{otherwise.} \end{cases}$
- When $t > 1$, $f_t(u) = 1$ if and only if
 1. $f_{t-1}(u) = 1$ or
 2. in the subgraph induced by $\{v : f_{t-1}(v) = 1\} \cup \{u\}$, the size of the connected component that contains u is at least $\theta(u)$.

²There are technologies that allows IPv6 messages to be *tunnelled* from one disconnected IPv6-enabled component of the network to another. However, tunnelling often incurs unacceptable performance penalties [19, 16], so we may think of utility as the number of IPv6-enabled destinations a node can be reach without tunnelling. (The other technologies we mention do not support tunnelling.)

Thus, $f_t(u)$ is u 's status as the t -th timestep; when $f_t(u) = 1$, we say u is turned on or is activated at the t -th timestep; when $f_t(u) = 0$, we say u is off or is inactive at the t -th timestep.

Definition 2.2 (Technology diffusion optimization problem.). *We say $\mathcal{T} = \min\{t : f_t = f_{t+1}\}$ is the completion time of the diffusion process. We say S is a feasible seedset with respect to $\{G, \theta\}$ if $\{v : f_{\mathcal{T}}(v) = 1\} = V$, i.e., all nodes are turned on by the process' completion time. Then, the technology diffusion optimization problem is to find the smallest feasible seedset S when G and θ are given as input.*

	\min	$\sum_{i \leq n} \sum_{t < \theta_i} x_{i,t}$	
subject to:	$\forall t, i :$	$x_{i,t} \in \{0, 1\}$	
	$\forall i :$	$\sum_{t \leq n} x_{i,t} = 1$	(permutation constraints)
	$\forall t :$	$\sum_{i \leq n} x_{i,t} = 1$	(permutation constraints)
	$\forall t > 1, i :$	$\sum_{\{v_i, v_{i'} \in E\}} \sum_{t' < t} x_{i',t'} \geq x_{i,t}$	(connectivity constraints)

Figure 1: Simple IP for the technology diffusion optimization problem.

3 Approximation Algorithm

We start with a detailed overview of key technical ideas in Section 3.1. We then describe our IP formulation in Section 3.2, and prove its correctness in Section 3.3; see also Appendix A for a discussion of the integrality gap that could arise in an alternate (simpler) version of the IP. Our rounding algorithm is presented in Section 3.4, and its correctness is proved in Section 3.5. Finally, to assist with the exposition, we present series of examples in Appendix B to illustrate constructions used by our algorithm.

3.1 Highlights of our algorithm

3.1.1 Linearization & formulating the IP

A major complication of our setting is that a node's activation decisions can depend on remote nodes in the graph. Consider a step in the diffusion process where there are multiple disconnected active components, and at time t , a single node u activates and joins these components into single 'giant' active component. This event would dramatically change the utility of nodes that are distant from node u but adjacent to new giant active component. These dramatic, non-local changes make it difficult to encode the problem as an IP; for instance, consider a natural IP formulation that uses indicator variables $y_{i,t}$ that are set iff node u_i is activated at timestep t . The IP would need to decide if the size of the active components including u_i at timestep t exceeds $\theta(u_i)$, a task that would likely require the use of threshold gates. This complicates matters since IPs are generally unable to express threshold gates.

It turns out that we can avoid threshold gates if the IP is only required to give an 2-approximation to the optimal seedset. To do this, we introduce the following notion:

Activation sequences. Given a seedset S , we can define an *activation sequence* T as a permutation from V to $[n]$ that indicates the order in which nodes activate. While Definition 2.1

imposes a one-to-one relationship between nodes and the timestep in which they activate, this notion of activation sequence is looser. Here, a seed may activate at any timestep, and a non-seed node u may activate at a timestep $T(u)$ as long as u is part of a connected component of size at least $\theta(u)$ in the subgraph induced by $\{u\} \cup \{v : T(v) < T(u)\}$.

Notice that we uniquely recover a feasible seedset S from an activation sequence T by deciding that node u is a seed iff $\theta(u) > T(u)$. Thus, we our IP will encode an activation function T , as a proxy for the seedset S .

We can use activation sequences to convert node’s activation decision from global (*i.e.*, influenced by remote nodes) to local (*i.e.*, influenced only by neighbors). To do this, we restrict the search space of our IP to *connected activation sequences*, *i.e.*, activation sequences T such that at every timestep t , the set of active nodes induces a connected subgraph of G . This way, we know that there are exactly t active and connected nodes at every timestep t , so any inactive node u can decide to activate if (a) at least one of its neighbors are active, and (b) the current timestep t is $t \geq \theta(u)$. Armed with these observations, encoding the IP becomes straightforward. See Figure 1:

Simple IP encoding. Let $x_{i,t}$ be the indicator variable such that $x_{i,t} = 1$ if and only if $T(v_i) = t$. The *permutation constraints* guarantee that the variables $x_{i,t}$ represent a permutation. The *connectivity constraints* ensure that if $x_{i,t} = 1$ (*i.e.*, node u_i activates at step t), there is some other node $u_{i'}$ such that $u_{i'}$ (a) is a neighbor of node u_i and (b) activates at earlier time $t' < t$. Finally, the objective function minimizes the size of the seedset by counting the number of $x_{i,t} = 1$ such that $t < \theta(u_i)$.

Bounding the size of the seedset. To see why restricting our search space to connected activation sequences results in a 2-approximation, consider a timestep when two or more disconnected active components merge into a single component, and notice that whenever this happens, there is exactly one *connector* node that activates and joins these two components. It turns out the if we add every connector node to the optimal seedset, we can rearrange the activation sequence to enforce connectivity. Since every connector node causes a decreases the number of disconnected components, and number of disconnected component is bounded size of the optimal seedset, we have the following lemma (proved in Appendix C):

Lemma 3.1. *The smallest seedset that can induce a connected activation sequence is at most twice the size of optimal seedset.*

3.1.2 Network flows & randomized rounding

Unfortunately, we can’t use the simple IP of Figure 1 to design our approximation algorithm, as it may exhibit a large integrality gap (see Appendix A). To deal with this, we need a new idea for our rounding approach: we shall simultaneously interpret fractional values returned by the LP both as *network flows*, and as probabilities.

The diffusion process as network flows. When a node u activates at time $T(u)$, we imagine a unit flow that originates at a seed node, and flows to node u along the network induced by the nodes activated prior to timestep $T(u)$. Our IP encodes this via *flow constraints*, that serve two purposes. First, they eliminate the pathological example of Appendix A. Second, they force our LP to return fractional flows $f \in [0, 1]$ that have the following pleasant interpretation: if there is a flow $f \in [0, 1]$ from a seed node to a node u at time t , then node u has probability f of activating at time t .

Fractional flows as probability mass. Suppose that at time t , there are two disjoint flows f_1 and f_2 originating from different seeds, and arriving simultaneously at node u . The total flow at node u at time t is then $f_1 + f_2$. What does this merge of two disjoint flows mean in our probabilistic interpretation? It turns out that the natural interpretation is already pretty sensible: with probability f_1 , the technology is diffused via the first flow, and with probability f_2 the technology is diffused via the second flow. Now, the probability that the technology is diffused to u via either of these two flows is $1 - (1 - f_1)(1 - f_2)$. If f_1, f_2 are both small, this probability becomes $\approx f_1 + f_2$, so that the total flow can be used to determine node u 's activation probability. On the other hand, if f_1 or f_2 is large, we are fairly confident that u should activate prior to time t , and so we can simply decide that $T(u) \leq t$ without incurring a large increase in the size of the seedset.

Connecting the seedset. Since network flows must originate at seed nodes, our rounding algorithm will require all seed nodes to activate before the non-seed nodes. Coupling this with the requirement that our activation sequence is connected, it follows that our rounding algorithm will return a *connected seedset* S (*i.e.*, the nodes in S induce a connected subgraph of G). To guarantee a connected seedset, our rounding algorithm samples candidate seed nodes and *glues* them together as follows:

GLUE-SEEDS(S)

- 1 **while** S is not connected
- 2 **do** Let C be the largest connected component in the subgraph induced by S .
- 3 Pick $u \notin C$. Let P be the shortest path connecting u and C in G .
- 4 Add nodes in P to S .
- 5 **return** S .

If r is the diameter of the graph (*i.e.*, the length of the longest shortest path in G), then gluing incurs a factor of $O(r)$ increase in the size of the seedset, which we show is optimal in Lemma 4.2.

min	$\sum_{i \leq n} \sum_{t < \theta_i} x_{i,t}$	
subject to:		
$\forall i, t :$	$x_{i,t} \in \{0, 1\}$	
$\forall i$	$\sum_{t \leq n} x_{i,t} = 1$	(permut'n constraints)
$\forall t$	$\sum_{i \leq n} x_{i,t} = 1$	(permut'n constraints)
$\forall \{i', i\} \notin E(G), t', t \in [n] :$	$e_{i', i, t', t} = 0$	
$\forall \{i', i\} \in E(G), t' \geq t :$	$e_{i', i, t', t} = 0$	
$\forall \{i', i\} \in E(G), t' < t :$	$e_{i', i, t', t} \in \{0, 1\}$	
$\forall i, t :$	$\sum_{t' < t} \sum_{\{i', i\} \in E} e_{i', i, t', t} = x_{i,t}$	(tree constraints)
$\forall i', t', t :$	$\sum_{\{i', i\} \in E} e_{i', i, t', t} \leq x_{i', t'}$	(activity constraints)
	$x_{1,1} = 1$	(make $X_{1,1}$ the source)
$\forall i, t \geq \theta(u_i) \forall$ partitions of $V(\mathbb{H}_t(i))$		
$S, \bar{S}, \text{s.t. } X_{1,1}^+ \in S, \text{ sk} \in \bar{S}$	$\sum_{e \in \delta(S, \bar{S})} c(e) \geq \sum_{\theta(u_i) \leq t' \leq t} x_{i,t'}$	(flow constraints).

Figure 2: Integer program for solving the technology diffusion problem.

3.2 Integer program

In Figure 2, we present the IP we used to design our approximation algorithm. We replace the simple connectivity constraints of the IP in Figure 1 with a more robust set of constraints that use network flows to enforce *connectivity*, i.e., that all active nodes are connected, and that node u activates after there are at least $\theta(u) - 1$ active nodes such that one of the active nodes is u 's neighbor.

Network flows. We require that when $x_{i,t} = 1$, we can push a flow of unit capacity from an arbitrary seedset node to u_i using only the subgraph induced by nodes activated before time t . To do this, we first extract the subgraph containing all the activate nodes at time t , and then ensure this subgraph admits a unit capacity of flow from a seed node to node u_i . We can do this by tracing the “trajectory” of the diffusion process, since when the seedset is connected, the technology diffusion process can be viewed as a growing tree. (See the example in Figure 6 of Appendix B.) We therefore impose our flow constraints over the following tree:

Timestamped diffusion tree. Node $T^{-1}(1)$ is the root. For each $t > 1$, the tree consists of all the nodes that are activated on or before time t . Moving from $t - 1$ -th to t -th step, the node $u = T^{-1}(t)$ is appended to the tree as a new leaf by adding a single directed edge from u to an arbitrary tree node v such that $v \in \{T^{-1}(1), \dots, T^{-1}(t - 1)\}$ and edge (u, v) is in the graph G . Finally, tree edge $(T^{-1}(t'), T^{-1}(t))$ is labeled with timestamps (t', t) . To encode the timestamped diffusion tree in our IP, we use *edge variables* $e_{i', i, t', t}$ such that $e_{i', i, t', t} = 1$ iff edge $\{i', i\}$ is in the timestamped diffusion tree with label (t', t) . The *tree constraints* ensure that each node in the timestamped diffusion tree has exactly one incoming edge, while the *activity constraints* ensure that only active nodes may have children in the tree.

To impose the flow constraints, we use the following hypergraph \mathcal{H} (see the example in Figure 7 of Appendix B):

The hypergraph \mathcal{H} . \mathcal{H} has vertex set $\{X_{i,t} : i, t \in [n]\}$, where vertex $X_{i,t}$ has “mass” $x_{i,t}$. For every non-zero $e_{i', i, t', t}$, \mathcal{H} has a directed edge from $X_{i', t'}$ to $X_{i,t}$ with capacity $e_{i', i, t', t}$. We also call vertices $\{X_{i,t} : t = \theta(u_i)\}$ the *threshold vertices* of \mathcal{H} , and let the *threshold line* be a line that joins all threshold vertices. All the vertices $X_{i,t}$ such that $t < \theta(u_i)$ are referred to as vertices *to the left of the threshold line*; all the rest are vertices *to the right of the threshold line*.

We think of the vertices to the left of the threshold line in \mathcal{H} as corresponding to seed nodes, while those to the right correspond to non-seed nodes. We want to ensure that all $X_{i,t}$ to the right of the threshold line (corresponding to non-seeds) have mass $x_{i,t}$ that exceeds neither (a) the capacity of the edge variables $e_{i', i, t', t}$, nor (b) the mass at the nodes that induced node i' to activate, e.g., $x_{i,t}$. To do this, we define a family of multi-flow problems as follows:

The (i, t) -flow problem. Fix an arbitrary i and t such that $t \geq \theta(u_i)$. Let j be the node corresponding to root of the timestamped diffusion tree, i.e., j activates at the first timestep so that $x_{j,1} = 1$. Let \mathcal{H}_t be the subgraph of \mathcal{H} induced by the vertices $\{X_{i', t'} : i' \in [n], t' \leq t\}$, where the mass on each \mathcal{H}_t 's vertices is interpreted as its capacity, and the directed edge from $X_{i', t'}$ to $X_{i,t}$ has capacity $e_{i', i, t', t}$. The (i, t) -flow problem is a multiple-sink flow problem over \mathcal{H}_t where that the source is $X_{j,1}$ and the sinks are vertices to the right of the threshold line $X_{i, \theta(u_i)}, X_{i, \theta(u_i)+1}, \dots, X_{i,t}$. The demand for sink $X_{i,t}$ is $x_{i,t}$. The *flow constraints* ensure that there is a solution to the (i, t) -flow problem for each i and $t \in [n]$ such that $t > \theta(u_i)$.

Implementing the flow constraints. The (i, t) -flow constraints are enforced via max-flow-min-cut, i.e., by using the fact that the minimum cut between the source and sinks is the same as

the maximum flow. For each (i, t) -flow problem, a simple min-cut formulation requires:

1. knowledge of the source, so assume $X_{1,1}$ is known to be the source, (A simple, polynomial-time way to achieve this is by guessing; run the IP n times, relabeling a different node in the graph as u_1 for each run, and use the run that returns the smallest seedset. The subsequent discussion corresponds to this “correct” run.)
2. a single sink, so we introduce node sk to connects sinks $X_{i,\theta(u_i)}, \dots, X_{i,t}$,
3. capacities on edges only, so we consider a new hypergraph $\mathbb{H}_t(i)$ that identical to \mathcal{H}_t except that each node $X_{j,\tau}$ is replaced with two nodes $X_{j,\tau}^+, X_{j,\tau}^-$ connected by a directed edge of capacity $x_{j,\tau}$.

In each hypergraph $\mathbb{H}_t(i)$, we need to supply sk with demand $\sum_{\theta(u_i) \leq t' \leq t} x_{i,t'}$. Let S and \bar{S} be a partition of the vertices of $\mathbb{H}_t(i)$, where $X_{1,1}^+ \in S$ and $sk \in \bar{S}$. Let $\delta(S, \bar{S})$ be the set of cross edges from S to \bar{S} . Let $c(e)$ be the capacity of each edge e in $\mathbb{H}_t(i)$. The *flow constraints* require that the capacity of all cuts are at least as large as the demand at the sink node.

3.3 Correctness of the IP

We combine Lemma 3.1 with the following lemma to conclude that the IP in Figure 2 returns a seedset of size 2opt . (Strictly speaking, this holds only if when u_1 is a seed in an optimal connected activation sequence due the constraint “(make $X_{1,1}$ the source)”; when this doesn’t hold, the seedset has size at most $2\text{opt} + r$, but as we discussed above we can ignore these runs of the IP.)

Lemma 3.2. *The IP in Figure 2 returns a connected activation sequence.*

Proof. We show that (a) all activation sequences satisfying the constraints of the IP must be connected, and (b) all connected activation sequences must satisfy the constraints.

We start with the first item. We use induction over t to show that the constraints in Figure 2 *excluding the flow constraints* force the IP to consider only connected activation sequences: The base case where $t = 1$ is trivial because there is only one activated node, namely u_1 . For the induction step, suppose the set of active nodes is connected up to time t , and $x_{i,t+1} = 1$ for some i . The tree constraint ensures that there exists an i' and a $t' < t + 1$ such that $e_{i',i,t',t+1} = 1$. The activation constraint also ensures that $x_{i',t'} \geq e_{i',i,t',t+1} = 1$. Therefore, there exists a node i' that is activated before time $t + 1$ and is connected to i , so that the set of active nodes are also connected at time $t + 1$.

To show the second item, we show that the flow constraints do not rule out any connected activation sequences. To do this, fix arbitrary i and t and consider two cases:

Case 1. If for all $t' \leq t$, $x_{i,t'} = 0$, then there is no demand for any of the sinks in the (i, t) flow problem, so the flow constraints trivially cannot rule out any solutions.

Case 2. Suppose there exists exactly one $t' \leq t$ where $x_{i,t'} = 1$. From the definition of the (i, t) flow problem, we must have $t' \geq \theta(u_i)$. Since the IP only searches through connected activation sequences T , it follows that T can be associated with timestamped diffusion tree that has a path from u_1 (the first node to activate) to u_i . It follows that there must also be a unit flow from $X_{1,1}$ to $X_{i,t'}$ in \mathcal{H} , and so this flow is a solution for the (i, t) -flow problem. \square

3.4 Relaxing and rounding algorithm

Our next task is to relax the IP in Figure 2 in the usual way³ (replacing the indicator variables with real variables over the $[0, 1]$ interval), and rounding the solution. Let σ be an optimal solution for the LP. Given σ , our rounding procedure will first reconstruct both the seedset S and an activation function T , and then iteratively reconcile inconsistencies between S and T .

3.4.1 Properties of the rounding procedure.

When we work with the relaxed LP, we also relax our notion of activation sequences. Now, we no longer assume that T is a permutation, and instead allow more than one node (or even no nodes at all) to activate in a single timestep. However, the following three properties ensure that this relaxed notion does not create problems:

- X1** (*Consistency*): After rounding is complete, T and S will be *consistent*; that is, T encodes an order of activation for a diffusion process induced by $\{G, \theta, S\}$ where any seed node $u \in S$ is allowed to activate at any time, and any non-seed node $u \notin S$ is allowed to activate at any time it is connected to an active component of size at least $\theta(u) - 1$.
- X2** (*Connectivity*): The activation sequence T is such that the set of active nodes is connected at all times (*i.e.*, $\bigcup_{t' \leq t} T^{-1}(t')$ is connected in G for $1 \leq t \leq n$).
- X3** (*Feasibility*): The activation sequence T is such that every node eventually activates (*i.e.*, $T(u) \leq n$ for each $u \in V$).

Thus, if the seedset S is consistent with the activation sequence T , then the seedset S is also feasible.

3.4.2 Overview of the the rounding procedure

Our rounding procedure works as follows. (To assist with the exposition, we also present an example of this rounding procedure in Appendix B.)

Reconstructing the seedset S . We use a randomized procedure to place a graph node u_i in the seedset with probability proportional to its *cumulative* mass to the left of the threshold line *i.e.*, $\sum_{t < \theta(u_i)} x_{i,t}$. This allows us to reconstruct a “small” seedset, but is not sufficient to guarantee that reconstructed seedset is feasible.

Reconstructing the activation function T . On the other hand, our reconstruction of the activation function T will guarantee that T is connected and feasible, but not that T encodes a “small” seedset. We do this by relying heavily on the finer information provided by the individual “mass” $x_{i,t}$, and interpreting these values as both network flows and probabilities. On one hand, we use the flow interpretation to construct the activation sequence as a function of (a) the mass of the hypergraph nodes \mathcal{H} to the left of the threshold line (*i.e.*, which will act as the source of the network flows) and (b) the structure of hypergraph \mathcal{H} (which will act as the network carrying these flows). On the other hand, we use the probability interpretation to ensure that distribution of $T(u_i)$ is, in expectation,

³Note that the relaxed LP contains an exponential number of constraints (namely, the flow constraints). Nevertheless, we can use the ellipsoid method to find an optimal solution in polynomial time using a separation oracle [31] that validates if each of the (i, t) -flow problems over \mathcal{H} have solutions, and if not, returns a min-cut constraint that is violated. This oracle can be constructed using algorithms in, *e.g.*, [17].

approximately characterized by the vector $(x_{i,1}, x_{i,2}, \dots, x_{i,n})$. Our reconstruction procedure will guarantee that T is connected and feasible, but not that T encodes a “small” seedset.

Reconciliation. We need to worry about situations where the activation function has fewer than t nodes active at timestep t (inclusive). If we ignore this, T and S could be inconsistent: T might suggest that a non-seed node $u \notin S$ activates “too early”, *i.e.*, at time $T(u)$ where there are fewer than $\theta(u)$ active nodes. To deal with this, we reconcile the reconstructed seedset S and reconstructed activation function T , so that they become consistent, and we have a seedset that is both feasible and small. We use an iterative, ℓ -step procedure, where ℓ is the number of possible threshold values we have in the problem instance, *i.e.*, $\theta : V \rightarrow \{\theta_1, \dots, \theta_\ell\}$. In each step, we again use the probability/network flow interpretation of our problem to “repair” situations when the activation sequence T has fewer than θ_j nodes activated at timestep θ_j by adding extra nodes to the seedset S .

Organization. To execute the above, we first construct a “preliminary” seedset S_0 and activation function T_0 (Section 3.4.3). In the reconciliation stage (Section 3.4.4), we iteratively construct a sequence of pairs $\{S_1, T_1\}, \dots, \{S_\ell, T_\ell\}$, so that at the end S_ℓ is a feasible solution and T_ℓ is consistent with S_ℓ (as proved in Section 3.5).

3.4.3 Reconstructing the preliminary seedset S_0 and activation function T_0

The following describes the process for obtaining the preliminary seedset S_0 and activation function T_0 from the fractional solution to the LP relaxation σ . We use a *randomized* process to obtain the preliminary seedset S_0 . Let $\epsilon > 0$ be an arbitrarily small number, which controls the tradeoff between running time and the size of seedset:

PRELIM-SEEDSET(\mathcal{H})

- 1 Initialize $S_0 \leftarrow \emptyset$.
- 2 For each node $u_i \in V$, add u_i to S_0 with probability $\min \left\{ 1, 24(1 + \epsilon) \ln(2n) \cdot \sum_{t < \theta(u_i)} x_{i,t} \right\}$.
- 3 Let $S_0 \leftarrow \text{GLUE-SEEDS}(S_0)$.
- 4 **return** S_0 .

(See Section 3.1 for the GLUE-SEEDS procedure.) We then *deterministically* obtain the activation sequence T_0 .

GET-SEQ(\mathcal{H}, S_0)

- 1 Initialize by flagging each $X_{i,t} \in \mathcal{H}$ as “inactive” by setting $b_{i,t} \leftarrow 0$.
- 2 $\forall u_i \in S_0, b_{i,t} \leftarrow 1$ for $t < \theta(u_i)$. // “Activate” each $X_{i,t}$ to the left of the threshold line
- 3 **for** $t \leftarrow 1$ to n
- 4 **do** $\forall i$:
- 5 **if** $(\exists i', t' : ((X_{i',t'}, X_{i,t}) \in E(\mathcal{H})) \wedge (b_{i',t'} = 1))$
- 6 $b_{i,t} \leftarrow 1$ for $t \geq \theta(u_i)$ // “Activate” each $X_{i,t}$ to the right of the threshold line
- 7 Obtain T_0 by taking $T_0(u_i) \leftarrow \min \{t : b_{i,t} = 1\}$.
- 8 **return** T_0 .

Notice that it is possible that T_0 is infeasible, *i.e.*, that T_0 is such that some node u never activates (denoted by $T_0(u) = \infty$). (See for example the first failure mode in Appendix B.) Thus, we repeat

PRELIM-SEEDSET(\mathcal{H}) with fresh randomness until we obtain S_0 that satisfies the two properties below. In Theorem 3.3 we show that a small number repetitions suffice to find S_0 that satisfies:

(P.1) Let $T_0 \leftarrow \text{GET-SEQ}(\mathcal{H}, S_0)$. For all i, t with $\sum_{t' \leq t} x_{i,t'} \geq \frac{1}{12(1+\epsilon)}$, then $T_0(u_i) \leq t$.

(P.2) The size of S_0 is at most

$$|S_0| = 24(1 + \epsilon)^2 \ln(2n)r \cdot (2\text{opt}) \quad (1)$$

Note that (P.1) immediately implies that the activation sequence T_0 is feasible; set $t = n$ in (P.1), so that $\forall i, \sum_{t' \leq n} x_{i,t'} = 1 \geq \frac{1}{12(1+\epsilon)}$, so $T_0(u_i) \leq n$.

3.4.4 Reconciliation procedure

The reconciliation procedure takes in the inconsistent preliminary seedset S_0 and activation function T_0 , and uses an ℓ -stage process to reconcile them. (Recall that ℓ defines the number of possible thresholds in our problem, *i.e.*, $\theta : V \rightarrow \{\theta_1, \dots, \theta_\ell\}$.)

We do this iteratively. At the k th stage of the reconciliation procedure (for all $k \in \{1, \dots, \ell\}$), we assume that seedset S_{k-1} and activation function T_{k-1} from the previous stage are “good” up to timestep θ_{k-1} , and use these to produce S_k and T_k that are “good” up to timestep θ_k . Our notion of “goodness up to time θ_k ” for seedset S_k and activation function T_k is defined as follows:

(C.1) S_k and T_k are *partially consistent up to time θ_k* (inclusive). That is, for any node u such that $T_k(u) \leq \theta_k - 1$ (a) either $u \in S_k$ (*i.e.*, u is a seed), or (b) in the subgraph induced by u and the set of nodes that are active up to time $\theta_k - 1$ according to T_k , the connected component containing u has size at least $\theta(u)$.

(C.2) T_k is such that the number of active nodes at time $\theta_j - 1$ (inclusive) is at least $\theta_j - 1$ for every $j \leq k$.

(C.3) S_k grows by an additive factor of at most

$$|S_k \setminus S_{k-1}| = r \max\{\log(2n/\epsilon), 24(1 + \epsilon) \cdot (2\text{opt})\} \quad (2)$$

Thus, the k th stage of the reconciliation procedure takes seedset S_{k-1} and produces a new seedset $S_k \supseteq S_{k-1}$, as follows:

UPDATE-SEEDSET(\mathcal{H}, S)

- 1 For each non-seed node $u_i \in V \setminus S$, add u_i to S with probability $\min\left\{1, 4(1 + \epsilon) \cdot \sum_{t < \theta(u_i)} x_{i,t}\right\}$.
- 2 Let $S \leftarrow \text{GLUE-SEEDS}(S)$.
- 3 **return** S .

Using $S_k \leftarrow \text{UPDATE-SEEDSET}(\mathcal{H}, S_{k-1})$, we can obtain a new activation sequence as $T_k \leftarrow \text{GET-SEQ}(\mathcal{H}, S_k)$. As before, we repeat UPDATE-SEEDSET(\mathcal{H}, S_{k-1}) with fresh randomness until we obtain S_k and T_k that satisfy (C.1) - (C.3). Theorem 3.3 shows that a small number of repetitions suffice.

3.5 Correctness of the approximation algorithm

Finally, we prove the correctness of our approximation algorithm.

Theorem 3.3. *Our algorithm outputs a feasible seedset of size at most*

$$24(1 + \epsilon)^2 \ln(2n)r \cdot (2\text{opt}) + \ell \cdot r \max\{\ln(2.89n/\epsilon), 24(1 + \epsilon)(2\text{opt})\} \quad (3)$$

by repeating PRELIM-SEEDSET at most $\tilde{O}(1)$ times and UPDATE-SEEDSET at most $\tilde{O}(n\ell/\epsilon)$ times.

Proof. Our proof proceeds by showing the preliminary seedset S_0 is ‘small’, and preliminary activation sequence T_0 is feasible and connected. We then inductively prove the reconciliation procedure resolves inconsistencies between the seedset and activation sequence, by adding a small number of new seeds to the seedset; because the seedset is consistent with a feasible activation sequence, the seedset is also feasible, and the theorem follows.

Preliminary seedset S_0 and activation sequence T_0 . We start by showing that a small number of repetitions of PRELIM-SEEDSET suffice to obtain a “small” preliminary seedset S_0 of size $24(1 + \epsilon)^2 \ln(2n)r(2\text{opt})$ (i.e., the first term in equation (3)), and a preliminary activation sequence T_0 that is both feasible and connected. Let \mathcal{H} be the hypergraph obtained from the optimal solution to the relaxation of the IP in Figure 2. The following suffices to show we obtain a “small” preliminary seedset after few repetitions of PRELIM-SEEDSET:

Lemma 3.4. *A single trial of PRELIM-SEEDSET(\mathcal{H}) satisfies property (P.2) with probability $1 - o(1)$.*

We prove this as Lemma H.1 (Appendix H.1). The proof uses a Chernoff bound to show that PRELIM-SEEDSET selects at most $24(1 + \epsilon)^2 \ln(2n)|\sigma|$ seeds with high probability. The proof proceeds to argue that since the optimal solution to the LP σ has size at most (2opt) (from Theorem 3.2), and the GLUE-SEEDS procedure used in PRELIM-SEEDSET expands the seedset by a factor of at most r , (P.2) holds with high probability, so that the size of S_0 is bounded by the first term in equation (3).

We move on the preliminary activation sequence $T_0 \leftarrow \text{GET-SEQ}(\mathcal{H}, S_0)$. To show that T_0 is connected, we use the fact that S_0 is connected and following connectivity lemma (which follows by construction of GET-SEQ, as proved in Appendix H.3)

Lemma 3.5 (Connectivity). *If S is a connected seedset and T is such that $T \leftarrow \text{GET-SEQ}(\mathcal{H}, S)$, then T is connected.*

Finally, note that the feasibility of T_0 (i.e., $T_0(u_i) \leq n$ for all $u_i \in V$) follows from immediately from property (P.1): set $t = n$ in (P.1) so that $\sum_{t' \leq n} x_{i,t'} = 1 \geq \frac{1}{12(1+\epsilon)}$. Furthermore:

Lemma 3.6. *A single trial of PRELIM-SEEDSET(\mathcal{H}) satisfies property (P.1) with probability $\Omega(1)$.*

We provide a sketch of this more substantial argument here, while the full proof is in Appendix H.2 as Lemma H.2. The proof relies heavily on the idea that the $x_{i,t}$ can be thought of both a network flows and probabilities. Fix an arbitrary u_i and let $t(i)$ be the smallest integer such that $\sum_{t \leq t(i)} x_{i,t} \geq \frac{1}{12(1+\epsilon)}$. To simplify the exposition (this simplification is not used in our full proof), suppose that either (a) $x_{i,t} = 0$ for all $t \geq \theta(u_i)$, or (b) $x_{i,t} = 0$ for all $t < \theta(u_i)$. These represent the two extreme cases for our lemma; the intermediate cases can be dealt with algebraic manipulations that “interpolate” between these extreme cases.

For case (a) all the mass is to the left of the threshold line, and we use a Chernoff bound to show that u is a seed with high probability. The interesting part of the proof comes for case (b) when all mass is to the right of the threshold line. To address this, we look at the hypergraph \mathcal{H} , and find a set of hypergraph nodes R to the left of the threshold lines such that (i)

$$\sum_{X_{j,t} \in R} \sum_{t < \theta(j)} x_{j,t} \geq 1/(12(1 + \epsilon)). \quad (4)$$

and (ii) for each $X_{j,t} \in R$, if GET-SEQ was to flag $X_{j,t}$ as “active” (*i.e.*, $b_{j,t} = 1$) then u_i will be activated before time $t(i)$. We first algorithmically extract the set of hypergraph nodes R , via network flow ideas; namely, we extract them from a feasible flow of the $(i, t(i))$ -flow problem. Next, when (4) holds, we use probabilistic analysis to show that whp at least one u_j corresponding to $X_{j,t} \in R$ will be selected as a seed, *i.e.*, PRELIM-SEEDSET will return S_0 such that $u_j \in S_0$. It follows that $X_{j,t}$ will be flagged as active in GET-SEQ, and u_i will activate before time $t(i)$.

Reconciliation procedure. We show that each of the ℓ stages of the reconciliation procedure grows the seedset by the additive factor in equation (2), resulting in the second term in equation (3). Moreover, we show the procedure results in a consistent seedset and activation function.

We do this inductively. For $k \in \{1, \dots, \ell\}$, we show that given S_{k-1} and T_{k-1} that satisfy conditions (C.1)-(C.2), it suffices to repeat UPDATE-SEEDSET $\tilde{O}(n/\epsilon)$ times in order to obtain S_k and T_k that satisfy conditions (C.1)-(C.3). (For the base case, note that (C.1) and (C.2) hold for preliminary seedset S_0 and activation function T_0 if we define $\theta_0 = 0$). We start by showing that (C.3) holds, so that seedset growth is small.

Lemma 3.7 (Seedset growth). *For the randomized UPDATE-SEEDSET procedure for obtaining S_k from S_{k-1} we have*

$$\Pr[|S_k \setminus S_{k-1}| \geq r \max\{\log(2n/\epsilon), 24(1 + \epsilon)(2\text{opt})\}] \leq \frac{\epsilon}{2n}$$

While this proof uses similar Chernoff bounds as in the proof of Lemma 3.4, we include it here in full in order to explain the somewhat ‘unnatural’-looking terms in equation (3).

Proof. Let ΔS_k be the set of seed nodes selected during step 1 of UPDATE-SEEDSET (before gluing) and recall that we add u_i to ΔS_k with probability $\max\{1, 4(1 + \epsilon) \sum_{t < \theta(u_i)} x_{i,t}\}$. It suffices to bound $|\Delta S_k|$, since $|S_k \setminus S_{k-1}| \leq r|\Delta S_k|$ after gluing in step 2 of UPDATE-SEEDSET. Observe that

$$\mathbb{E}[|\Delta S_k|] = \sum_{i \leq n} \min\{1, 4(1 + \epsilon) \sum_{t < \theta(u_i)} x_{i,t}\} \leq 4(1 + \epsilon) \sum_{i \leq n} \sum_{t < \theta(u_i)} x_{i,t} \leq 4(1 + \epsilon)(2\text{opt}).$$

(where the last equality follows because the objective function of the LP has size at most (2opt)). Let $\delta = \max\{\log(2n/\epsilon), 24(1 + \epsilon)(2\text{opt})\}$. We bound the event that $|\Delta S_k| \geq \delta$ in two cases:

Case 1. $24(1 + \epsilon)(2\text{opt}) \geq \log(2n/\epsilon)$. Notice first that $\mathbb{E}[|\Delta S_k|] \leq 6 \times 4(1 + \epsilon)(2\text{opt})$ and we can apply the Chernoff bound (Part 2 of Theorem G.1):

$$\Pr[|\Delta S_k| \geq \delta] \leq \Pr[|\Delta S_k| \geq 24(1 + \epsilon)(2\text{opt})] \leq 2^{-24(1 + \epsilon)(2\text{opt})} \leq 2^{-\log(2n/\epsilon)} \leq \epsilon/2n.$$

Case 2. $24(1 + \epsilon)(2\text{opt}) \leq \log(2n/\epsilon)$. We now have $\mathbb{E}[|\Delta S_k|] \leq 6 \times 4(1 + \epsilon)(2\text{opt}) \leq \ln(2n/\epsilon)$. Using the same Chernoff bound:

$$\Pr[|\Delta S_k| \geq \delta] \leq \Pr[|\Delta S_k| \geq \log(2n/\epsilon)] \leq 2^{-\log(2n/\epsilon)} = \epsilon/2n.$$

□

Next, we have a simple lemma (Lemma H.6) that shows that partial consistency condition (C.1), is immediate given that S_k and T_k that satisfy condition (C.2), and S_{k-1} and T_{k-1} satisfy condition (C.1) and (C.2). We leave that lemma to Appendix H.4, and move on to our main task: showing that a single trial of UPDATE-SEEDSET produces S_k and T_k that satisfy (C.2) with probability $\tilde{O}(\epsilon/n)$. We do this in two steps, (with proofs in Appendix H.5 - H.6):

1. Ideally, we would like activation function T_{k-1} to have $\theta_k - 1$ nodes active by time $\theta_k - 1$ (inclusive) so that (C.2) will be met after T_{k-1} is updated to T_k . However, this may not be the case for T_{k-1} . Thus, we compute the gap between $\theta_k - 1$ and the number of active nodes at time $\theta_k - 1$ according to activation function T_{k-1} and show such gap can be “filled” whp after we execute UPDATE-SEEDSET once. The following lemma, proved in Appendix H.5 follows almost immediately from algebra:

Lemma 3.8 (Gap size.). *At beginning of the k -th stage, define the “gap” as*

$$\rho = \theta_k - 1 - |\{u_i : T_{k-1}(u_i) \leq \theta_k - 1\}| \quad (5)$$

Note that γ is the total mass in \mathcal{H} to the left of the threshold line corresponding the non-seed nodes $u_i \in V \setminus S_{k-1}$. It follows that

$$\rho \leq \gamma = \sum_{u_i : T_{k-1}(u_i) \geq \theta_k} \sum_{t < \theta_k} x_{i,t}$$

It follows that $\rho \leq \gamma \triangleq \sum_{u_i : T_{k-1}(u_i) \geq \theta_k} \sum_{t < \theta_k} x_{i,t}$.

2. Next, we show that the number of nodes moved to a timestep earlier than θ_k in T_k , is larger than the gap ρ with probability at least ϵ/n . Thus, it follows that condition (C.2) holds with probability at least ϵ/n .

Lemma 3.9. *Let*

$$\hat{\rho} = |\{u_i : (T_{k-1}(u_i) \geq \theta_k) \wedge (T_k(u_i) < \theta_k)\}|$$

be the number of nodes that are moved to a timestep earlier than θ_k in activation sequence T_k (relative to T_{k-1}). Then $\Pr[\hat{\rho} \geq \gamma] \geq \epsilon\gamma/n$.

This lemma carries that majority of the substance of this part of the proof. We again need to combine a network flow interpretation with probabilistic analysis in a manner that is similar, but more sophisticated, than the analysis for Lemma 3.6. The main complication is that in Lemma 3.6 the right-hand side of the inequality (4) is *constant*, while here we must bound the sum of the masses through the hypergraph nodes in R with a non-constant value that is related to the gap γ . The complete proof is presented in Appendix H.6.

Thus, after the reconciliation procedure, S_ℓ has size as in equation (3), and is consistent with activation sequence T_ℓ . Since T_ℓ is feasible, S_ℓ is also feasible and the theorem follows. $\square \quad \square$

4 Lower bounds

We present two lower bounds. First, we show an $\Omega(\log n)$ inapproximation lower bound for polynomial time algorithms, assuming no efficient $(1 - o(1))$ -approximation algorithm for set cover problem exists. This lower bound holds even when r (the diameter of the graph) and ℓ (threshold granularity) are constants, and implies that our algorithm is close to optimal for a wide range of interesting diffusion problems described in Section 1, where r and ℓ are $\tilde{O}(1)$:

threshold step length:	$c = 1$		$c = 5$		$c = 10$		$c = 20$	
	Size	Jaccard	Size	Jaccard	Size	Jaccard	Size	Jaccard
degree	11.8	0.42	20.9	0.36	24.45	0.38	41.75	0.46
degree-threshold	8.95	0.41	15.40	0.42	19.00	0.44	33.25	0.55
betweenness	10.50	0.45	19.65	0.39	24.2	0.38	40.85	0.47
degree discounted	11.2	0.39	21.55	0.34	25.35	0.36	41.60	0.45
degree connected	12.9	0.35	22.65	0.29	25.90	0.33	43.25	0.44
ip_solver	6.45	1	11.15	1	13.75	1	23.45	1
degree overlap		0.44		0.39		0.37		0.39
betweenness overlap		0.47		0.39		0.37		0.40

Table 1: Comparison the IP of Figure 1 to several heuristics.

Lemma 4.1. *There is no $c \ln n$ -approximation algorithm (for some constant c) for the technology diffusion optimization problem for a general graph, even if the seedset is required to be connected, and graph diameter r and threshold granularity ℓ are $O(1)$.*

The proof, in Appendix D, is a reduction that takes an α -approximation algorithm for the technology diffusion problem and returns an $O(\alpha)$ -approximation algorithm for set cover problem. The lemma follows because the set cover problem cannot be approximated within a factor of $\Theta(\ln n)$ (see [2] and references therein).

Second, we show an unconditional $\Omega(r)$ -inapproximation lower bound for the family of algorithms that only search for connected seedsets (*i.e.*, seedsets that induce a connected subgraph of G), that explains our approximation algorithm’s dependence on graph diameter r :

Lemma 4.2. *For any fixed integer r , there exists an instance of technology diffusion problem $\{G, \theta\}$ such that (a) the diameter of G is $\Theta(r)$, and (b) the optimal connected seedset is at least $\Omega(r)$ larger than the optimal seedset.*

The proof is in Appendix E. It follows that circumventing polynomial dependencies in graph diameter r requires algorithms that can return *disconnected seedsets*. As discussed in Section 3.1, we believe that doing this will require substantially different techniques, and is thus an interesting direction for future work.

5 Going beyond heuristics.

Given the prevalence of heuristics like “choose the high degree nodes” in the literature on technology diffusion in communication networks (*e.g.*, [6, 3, 14]), we sanity-check our approach against several heuristics. We emphasize that our goal in the following is to *give evidence* that we can find solutions that are substantially different from natural heuristics.

We considered problem instances where (a) $G(V, E)$ is 200-node preferential attachment graph with node outdegree randomly chosen from $\{1, 2, 3, 4\}$ [1], and (b) thresholds θ randomly chosen from $\{\max\{2, c\}, 2c, 3c, \dots, \lceil \frac{200}{c} \rceil \cdot c\}$. We ran four groups of experiments with threshold step-length parameter c fixed to 1, 5, 10, and 20 respectively. With each group, we used a fresh random preferential attachment graph, and repeated the experiment five times with a fresh random instance

of the threshold functions. We solve each of these 20 problem instance using the IP formulation presented in Figure 1 (with the extra restriction that the highest degree node must be part of the seedset) and the Gurobi IP solver. We compare the result against five heuristics that iteratively pick a node u with property X from the set of inactive nodes, add u to the seedset S' , activate u , let u activate as many nodes as possible, and repeats until all nodes are active. We instantiate property X as:

- (a) *degree*: highest degree,
- (b) *degree-threshold*: highest $(\text{degree}) \times (\text{threshold})$,
- (c) *betweenness*: highest betweenness centrality,
- (d) *degree discounted*: highest degree in the subgraph induced by the inactive nodes [8],
- (e) *degree connected*: highest degree and connected to the active nodes.

For each group, Table 1 presents the average seedset size and the average Jaccard index $\frac{|S \cap S'|}{|S \cup S'|}$ between IP seedset S the heuristic seedset S' . We also compute the fraction of nodes in S that are also part of the top- $|S|$ nodes in terms of (a) degree (the row denoted “degree overlap”), and (b) betweenness centrality (“betweenness overlap”). The results of Table 1 do indeed give evidence that our IP can return seedsets that are substantially different (and often better), than the seedsets found via heuristics.

Acknowledgements

We thank Nadia Heninger, Nicole Immorlica, Prasad Raghavendra, Jennifer Rexford and Santosh Vempala for discussions about earlier incarnations of this model, Michael Mitzenmacher and Michael Schapira for comments on this draft, and Boaz Barak, Phillipa Gill and David Karger for helpful suggestions.

References

- [1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74, Jan 2002.
- [2] N. Alon, D. Moshkovitz, and S. Safra. Algorithmic construction of sets for k -restrictions. *ACM Trans. Algorithms*, 2:153–177, Apr. 2006.
- [3] I. Avramopoulos, M. Suchara, and J. Rexford. How small groups can secure interdomain routing. Technical report, Princeton University Comp. Sci., 2007.
- [4] B. Barak, S. Goldberg, and D. Xiao. Protocols and lower bounds for failure localization in the Internet. In *IACR EUROCRYPT*, 2008.
- [5] B. Briscoe, A. Odlyzko, and B. Tilly. Metcalfe’s law is wrong. *IEEE Spectrum*, 2006.
- [6] H. Chang, D. Dash, A. Perrig, and H. Zhang. Modeling adoptability of secure BGP protocol. In *Sigcomm*, 2006.
- [7] N. Chen. On the approximability of influence social networks. In *ACM-SIAM Symposium on Discrete Algorithms*, 2008.
- [8] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proc. 15th Conference on Knowledge discovery and data mining, KDD '09*, pages 199–208. ACM, 2009.
- [9] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden. Tussle in cyberspace: defining tomorrow’s Internet. *Trans. on Networking*, 2005.

- [10] S. Deering and R. Hinden. RFC 2460: Internet Protocol, Version 6 (IPv6) Specification. <http://www.ietf.org/rfc/rfc2460.txt>, 1998.
- [11] P. Domingos and M. Richardson. Mining the network value of customers. In *Proc. 7th Conf on Knowledge discovery and data mining*, KDD '01, pages 57–66, New York, NY, USA, 2001. ACM.
- [12] B. Edelman. Running out of numbers: Scarcity of ip addresses and what to do about it. Technical report, Harvard Business School, 2009.
- [13] H. A. Elmore, L. J. Camp, and B. P. Stephens. Diffusion and adoption of ipv6 in the arin region. In *Workshop on the Economics of Internet Security*, 2008.
- [14] P. Gill, M. Schapira, and S. Goldberg. Let the market drive deployment: A strategy for transitioning to BGP security. *SIGCOMM'11*, 2011.
- [15] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, May 1978.
- [16] R. Guérin and K. Hosanagar. Fostering ipv6 migration through network quality differentials. *SIGCOMM Comput. Commun. Rev.*, 40:17–25, June 2010.
- [17] J. Hao and J. B. Orlin. A faster algorithm for finding the minimum cut in a graph. In *SODA*, 1992.
- [18] M. Howarth, P. Flegkas, G. Pavlou, N. Wang, P. Trimintzios, D. Griffin, J. Griem, M. Boucadair, P. Morand, H. Asgari, and P. Georgatsos. Provisioning for inter-domain quality of service: the MESCAL approach. *IEEE Communications Magazine*, June 2005.
- [19] G. Huston. Stacking it up: Experimental observations on the operation of dual stack services. In *NANOG'52*, 2011.
- [20] Y. Jin, S. Sen, R. Guerin, K. Hosanagar, and Z.-L. Zhang. Dynamics of competition between incumbent and emerging network technologies. *NetEcon*, 2008.
- [21] D. Joseph, N. Shetty, J. Chuang, and I. Stoica. Modeling the adoption of new network architectures. In *CoNEXT'07: Conference on emerging Networking EXperiments and Technologies*, 2007.
- [22] M. L. Katz and C. Shapiro. Technology adoption in the presence of network externalities. *Journal of Political Economy*, 94(4):822–41, August 1986.
- [23] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD*, 2003.
- [24] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol (S-BGP). *JSAC*, 2000.
- [25] M. Lepinski, editor. *BGPSEC Protocol Specification*. IETF Network Working Group, Internet-Draft, Mar. 2011. Available from <http://tools.ietf.org/html/draft-lepinski-bgpsec-protocol-00>.
- [26] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2005.
- [27] B. Metcalfe. Metcalfe's law: A network becomes more valuable as it reaches more users. *InfoWorld*, 1995.
- [28] S. Morris. Contagion. *The Review of Economic Studies*, Jan., 2000.
- [29] A. Ozment and S. E. Schechter. Bootstrapping the adoption of internet security protocols. In *The Fifth Workshop on the Economics of Information Security (WEIS 2006)*., 2006.
- [30] T. C. Schelling. *Micromotives and Macrobehavior*. Norton, 1978.
- [31] D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2010.

- [32] A. Yaar, A. Perrig, and D. Song. SIFF: a stateless internet flow filter to mitigate ddos flooding attacks. *IEEE Symposium on Security and Privacy*, 2004.

A The simple integer program

Our IP in Figure 2 is quite complex; in this section we show why we were not able to base our approximation algorithm on the simpler integer program of Figure 1. For the sake of exposition, we suppose that $x_{i,t}$ is a mass that gives a measure of the probability that node u_i activates at time t , and refer to the example in Figure 3:

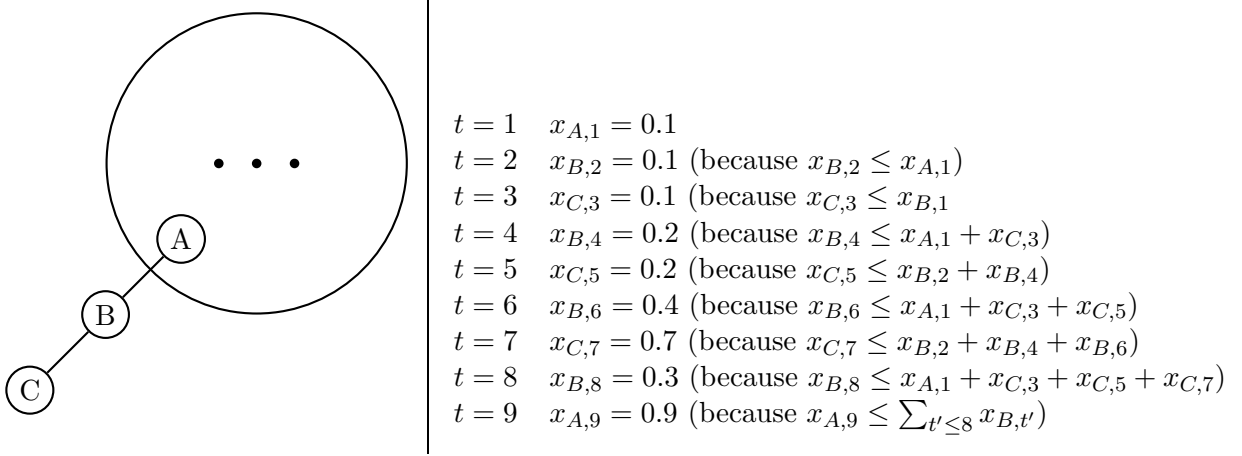


Figure 3: Here, the solution returned by the relaxed LP is unlikely to be helpful in rounding.

Figure 3: Suppose LP returns a solution such that at $t = 1$, node A has mass 0.1, while all other nodes have mass 0. The constraints repeatedly allow mass from node A to circulate through nodes B and C and then back to A. (See the right hand side of Figure 3 for the variable assignments over the time). Finally, at $t = 9$, enough mass has circulated back to A so that A will have mass 0.9, so that A has “probability” 0.9 of activating. Note that this is highly artificial, as all of this mass originated at A to begin with! In fact, no matter how we interpret these $x_{i,t}$, the example suggests that this “recirculation of mass” is not going to give us any useful information about when node A should actually activate.

We took care of this recirculation of mass by introducing the timestamped diffusion tree and the flow constraints, to prevent mass from circulating from a node back to itself at some future time (*e.g.*, From node A at time $t = 1$ back to itself at time $t = 9$). To illustrate how the flow constraints eliminate the pathological example of Figure 3, in Figure 4 we presents the hypergraph \mathcal{H} corresponding to the first 4 timesteps of activation sequence showed in Figure 3. Notice that \mathcal{H} violates the constraints of the IP in Figure 2 because the $(B, 4)$ -flow problem has in total demand 0.2 (*i.e.*, $x_{B,2} = 0.1$ and $x_{B,4} = 0.1$) but there is no way to supply this demand from $X_{A,1}$.

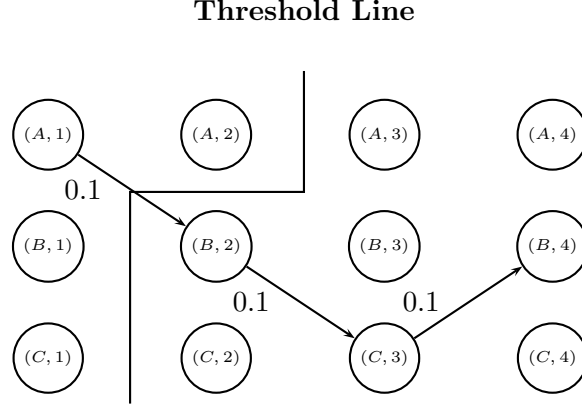


Figure 4: The hypergraph \mathcal{H} corresponding to the activation sequence of Figure 3.

B Some Examples

We now present a few examples of the constructions we used in our main IP in Figure 2, and its relaxation, as described in Section 3.4.

We start with a technology diffusion problem in Figure 5, and present an optimal timestamped diffusion tree and activation function T for this problem in Figure 6. The hypergraph \mathcal{H} that would result from an *integer solution* to the IP, is presented in Figure 7. Notice that the edges in \mathcal{H} form a tree corresponding to the timestamped diffusion tree in Figure 6. Meanwhile, the hypergraph \mathcal{H} in Figure 8, is constructed from a *fractional solution* of the LP for the problem in Figure 5. Notice that the edges in \mathcal{H} do not correspond to a tree (so that in the LP relaxation, we no longer have the assurance that \mathcal{H} encodes a timestamped diffusion tree); however, \mathcal{H} does not contain any violations of the flow constraints.

We now use \mathcal{H} in Figure 8 to illustrate our rounding procedure. In particular, we illustrate two “failure modes” where at intermediate stages of our algorithm, the seedset S and activation sequence T violate one of the three conditions in Section 3.4.2, as well as a “success mode” where S and T adhere to all three of these conditions.

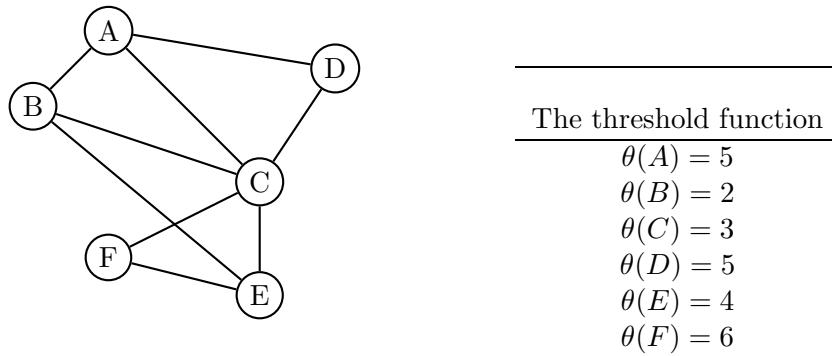


Figure 5: An instance of technology diffusion problem.

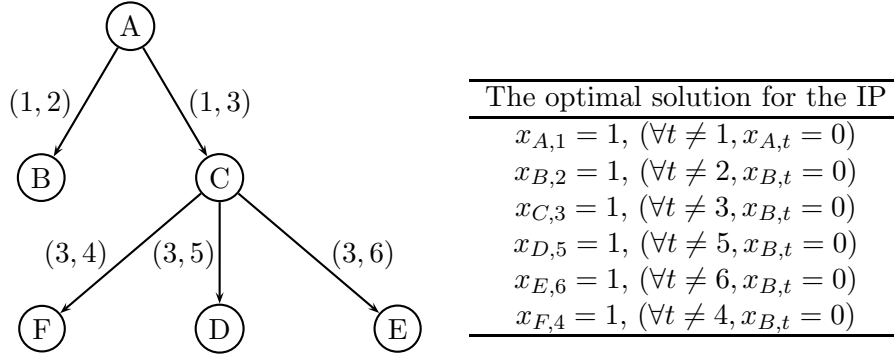


Figure 6: A solution for the technology diffusion problem in Figure 5 (on the right hand side) and the corresponding timestamped diffusion tree (on the left hand side).

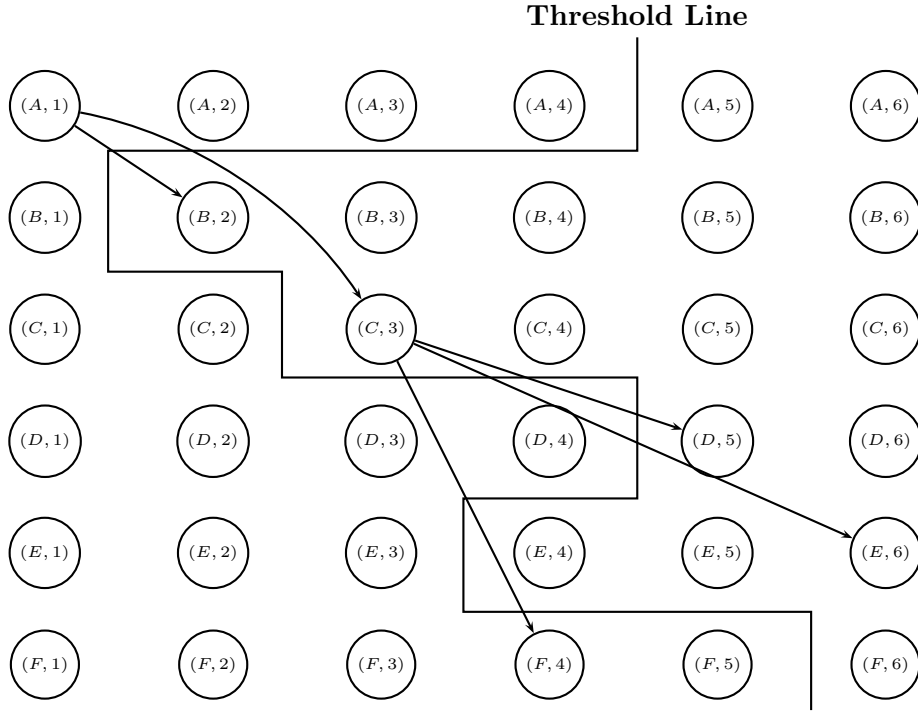


Figure 7: The \mathcal{H} graph corresponding to the diffusion tree in Figure 6.

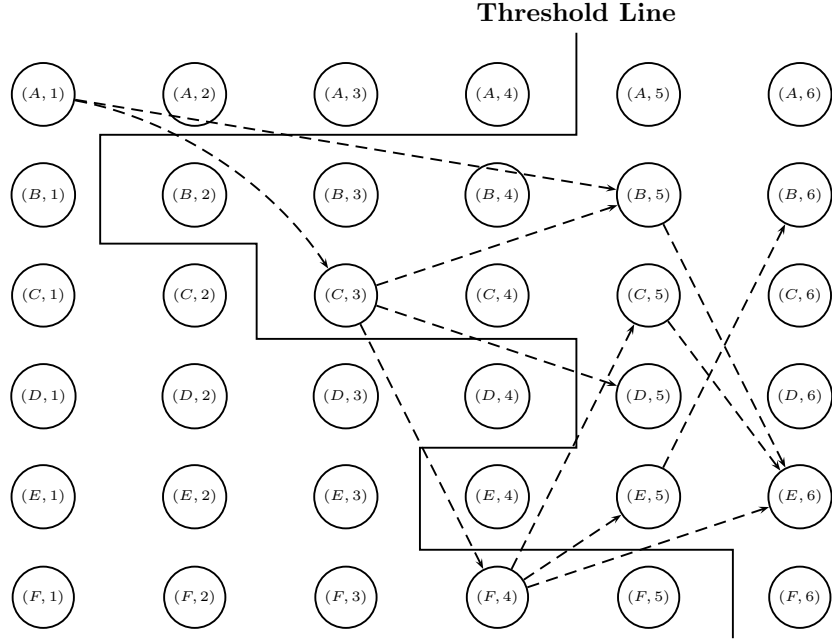


Figure 8: The \mathcal{H} graph obtained from the solution of the relaxed program Figure 6.

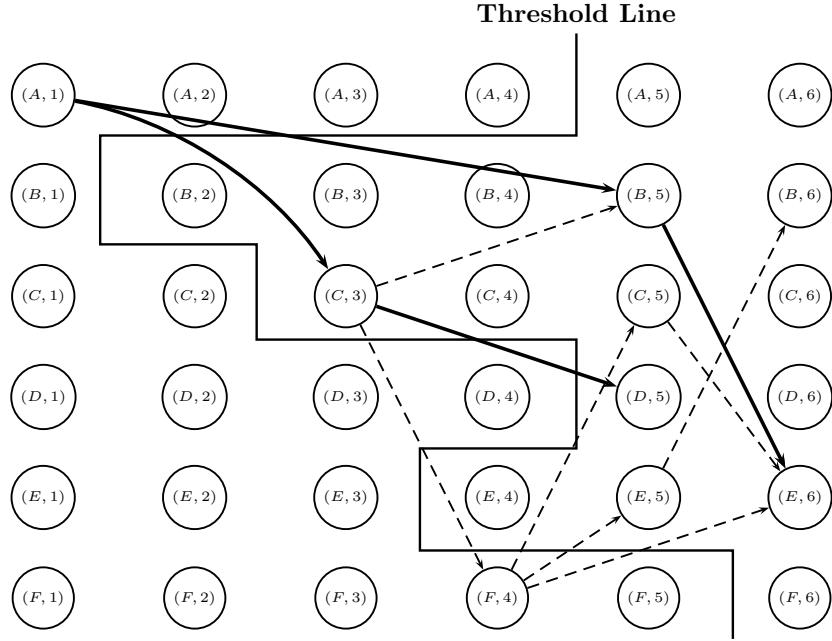


Figure 9: Setting the seed set $S = \{A\}$ and using the flow graph from \mathcal{H} , the flag variables are activated along the solid trajectories. We have $T(A) = 1$, $T(B) = 5$, $T(C) = 3$, $T(D) = 5$, $T(E) = 6$, $T(F) = \perp$.

Failure 1: Infeasible activation function. Let $S_0 \leftarrow \text{PRELIM-SEEDSET}(\mathcal{H})$ and $T_0 \leftarrow \text{GET-SEQ}(\mathcal{H}, S_0)$ for the hypergraph \mathcal{H} in Figure 8. Suppose $S_0 = \{A\}$. Figure 9 shows the update of the flag variables $b_{i,t}$ inside GET-SEQ using seedset $S_0 = \{A\}$. This example gives us

- $S_0 = \{A\}$.
- $T_0(A) = 1, T_0(B) = 5, T_0(C) = 3, T_0(D) = 5, T_0(E) = 6, T_0(F) = \perp$.
That is, $T_0 = (A, \perp, C, \perp, \{D, B\}, E)$

First, note the flag variables are activated along the solid trajectories, and that even though hypergraph node $(C, 3)$ is flagged as active, there is no solid trajectory from $(C, 3)$ and $(F, 4)$; this is because $F \notin S_0$ and GET-SEQ only activates nodes to the right of the threshold line. Moreover, observe that GET-SEQ can sometimes leave certain timesteps in the activation function T_0 empty, as with timesteps 2 and 4 in the example above. Finally, note that in this example, our activation function T_0 is *infeasible*; that is, node F never turns on! Thus, this example represents a failed run of PRELIM-SEEDSET that violates condition (P.1); so we must rollback PRELIM-SEEDSET and re-execute it until it returns a *feasible* activation sequence (*i.e.*, where all nodes eventually turn on).

Failure 2: Inconsistent preliminary seedset and activation function. Suppose now that we re-ran PRELIM-SEEDSET until it returned preliminary seedset $S_0 = \{A, F\}$. Returning to Figure 9, GET-SEQ would now flag $(F, 1), (F, 2), (F, 3), (F, 4), (F, 5)$ as active (because F is a seed), and we would have additional solid trajectories from hypergraph node $(F, 4)$ to hypergraph nodes $(C, 5), (E, 5)$ and $(E, 6)$. Since we consider nodes to be active at the timestep corresponding to their earliest active hypergraph node, our activation function would become:

$$T_0 = (\{A, F\}, \perp, C, \perp, \{B, D, E\}, \perp)$$

First, note that our activation sequence is now feasible – every node eventually turns on in T_0 – so that S_0 and T_0 would be accepted as a preliminary seedset and activation sequence. However, observe that S_0 and T_0 are *inconsistent* (as defined in Section 3.4.2). To see why, note that T_0 has node D activating at timestep 4. Referring to the threshold line, we note that D has threshold 4, and according to S_0 , we have that D is not a seed. However, T_0 indicates that just before timestep 4 there are only *three* nodes that are active (nodes A, F, C). This is precisely why S_0 and T_0 are inconsistent; because they suggest that a non-seed node prematurely activates, *i.e.*, when the number of active nodes is less than as required by his threshold!

This is where our reconciliation procedure comes in; as we discuss in Section 3.4.4, our reconciliation procedure iteratively adds additional nodes to the seedset until the resulting activation function and seedset become consistent. We note that in this example, it suffices for the reconciliation procedure to add either node B, D or F to the preliminary seedset.

Success: Feasible and consistent seedset and activation function. Finally, suppose the seedset becomes $\{A, B, F\}$. Then, the activation function becomes

$$T = (\{A, B, F\}, \perp, C, \perp, \{D, E\}, \perp)$$

which is the sort of activation function that we would like to have at the termination of the reconciliation procedure, since it is both feasible (every node eventually turns on) and consistent (non-seed nodes never activate prematurely).

C Optimal connected activation sequences provide a 2-approximation

Recall that a connected activation sequence T is such that the set of active nodes at any timestep t induces a connected subgraph of G , while a connected seedset is such that all nodes in S induce a connected subgraph of G . Notice that requiring the activation sequence T to be connected is *weaker* than requiring a connected seedset S : since T allows a seed to activate *after* a non-seed, the connectivity of T can be preserved by non-seeds whose activation time occurs *between* the activation times of the seed nodes.

We now show that the smallest seedset that gives rise to a feasible connected activation sequence is at most twice the size of the optimal seedset opt .

of Lemma 3.1. Given an optimal activation sequence T_{opt} and seedset opt , we shall transform it into a connected activation sequence T . Along the way, we add nodes to the seedset in manner that increases its size by a factor of at most 2.

Notation. Let $G_i(T)$ be the subgraph induced by the first i active nodes in T . We say a node u is a *connector* in some activation sequence T if the activation of u in T connects two or more disconnected components in $G_{T(u)-1}(T)$ into a single component.

Creating a connected activation sequence. Notice that an activation sequence $T(\cdot)$ is connected if and only if there exists no *connector* in the sequence. Thus, it suffices to iteratively “remove” connectors from T until no more connectors remain.

To do this, we initialize our iterative procedure by setting $T \leftarrow T_{\text{opt}}$. Each step of our procedure then finds the earliest connector u to activate in T , adds u to the seedset, and applies the following two transformations:

Transformation 1: First, we transform T so that every component in $G_{T(u)}(T)$ is directly connected to u . Let $D(u)$ be the subsequence of T such that every node in $D(u)$ both activates before u , and is part of a component in $G_{T(u)}(T)$ that is *not* connected to u . Transform T so the subsequence $D(u)$ appears immediately *after* node u activates. (This does not harm the feasibility of T , because the nodes in $D(u)$ are disconnected from the other nodes in $G_{T(u)}(T)$ that activate before u .)

Transformation 2: Next, we transform the activation sequence so that it is connected up to time $T(u)$. To see how this works, assume that there are only two connected components C_1 and C_2 in $G_{T(u)-1}(T)$, where $|C_1| \geq |C_2|$. Our transformation is as follows:

1. First, activate the nodes in C_1 as in $T(\cdot)$.
2. Then, activate u . (This does not harm feasibility because we added u to the seedset. Connectivity is ensured because u is directly connected to C_1 .)
3. Finally, have all the nodes in C_2 activate immediately after u ; the ordering of the activations of the nodes in C_2 may be arbitrary as long as it preserves connectivity. (This does not harm feasibility because (a) seed nodes may activate at any time, and (b) any non-seed $v \in C_2$ must have threshold $\theta(v) \leq |C_2| \leq |C_1|$ and our transformation ensures that at least $|C_1| + 1$ nodes are active before any node in C_2 activates.)

We can easily generalize this transformation to the case where k components are connected by u by letting $|C_1| \geq |C_2| \geq \dots \geq |C_k|$ and repeating step (3) $k - 1$ times. At this point the transformed activation sequence is feasible and connected up to time $t = 1 + |C_1| + |C_2| + \dots + |C_k|$.

Seedset growth. It remains to bound the growth of the seedset due to our iterative procedure. We do this in three steps. First, we observe that number of extra nodes we added to the seedset is bounded by the number of steps in our iterative procedure. Next, we iteratively apply the following claim (proved later) to argue that the number of steps in our iterative procedure is upper bounded by number of connectors in the optimal activation sequence, T_{opt} :

Claim C.1. *Let T_j be the activation sequence at the start of j^{th} step. The number of connectors in T_{j+1} is less than the number of connectors in T_j .*

Thus, it suffices to bound the number of connectors in T_{opt} . Our third and final step is to show that the number of connectors in T_{opt} is bounded by $|\text{opt}|$. To do this, we introduce a potential function $\Phi(t)$ that counts the number of disconnected components in $G_{T_{\text{opt}}(t)}(T)$, and argue the following:

- For every connector u that activates at time t in T_{opt} and joins two or more disconnected components, there is a corresponding decrement in Φ , *i.e.*, $\Phi(t) \leq \Phi(t-1) - 1$.
- Next, we have that $\Phi(1) = \Phi(|V|)$, since at the first timestep, there is only one active node, and at the last timestep all the nodes in the graph are active and form a single giant component. Thus, for every unit decrement in Φ at some time t , there is a corresponding unit increment in Φ at some other time t' .
- Finally, for any unit increment in Φ , *i.e.*, $\Phi(t') = \Phi(t'-1) + 1$, it follows that a new disconnected component appears in $G_{T_{\text{opt}}(t')}(T)$. This implies that a new seed activates at time t' . Thus, it follows that the number of unit decrements of Φ is upperbounded by the size of the seedset $|\text{opt}|$.

Thus, we may conclude the the number of connectors added to seedset in our iterative procedure is upperbounded by the number of connectors in T_{opt} which is upperbounded by the size of the optimal seedset opt , and the lemma follows. \square

The correctness of Claim C.1 is fairly intuitive, given that our transformations always preserve the ordering of the nodes that are not in the components joined by node u . We include the proof for completeness.

of Claim C.1. We make use of the following observation:

Observation 1: If two activation sequences T and T' have a common suffix, *i.e.*, $T = T'$ for timesteps $\tau, \tau+1, \dots, |V|$, then T and T' contain the same number of connectors after time $\tau-1$.

Let $t = T_j(u)$. By construction, no connectors exist in T_j prior to time t . Furthermore, we can use Observation 1 to argue that T_j and T_{j+1} contain the same number of connectors after time t . Thus, it suffices to show that Transformations 1 and 2 in the j^{th} step of our iterative procedure do not introduce new connectors that activate in prior to time t .

Let T^* be the activation sequence after Transformation 1 in the j^{th} step of our iterative procedure, and let $t' = T^*(u)$. We can see that (1) no new connectors activate before time t' in T^* (since, before t' our construction ensures that T^* consists only of disconnected active components that are joined by u) and (2) no new connectors activate between time $t'+1$ and t inclusive (since (a) u was chosen as the earliest connector in T_j , and (b) Transformation 1 preserves the order of the nodes that activate between time $t'+1$ and t inclusive in T^*).

Finally, we conclude by arguing that Transformation 2 cannot introduce new connectors by (1) applying Observation 1 to the nodes after t' and (2) observing that after Transformation 2, the nodes that activate before t' create a single connected component, and thus by definition cannot contain any connectors. \square

D Reduction to Set Cover

Let us recall the definition (of the optimization version) of the set cover problem: given a finite universe \mathcal{U} and a family \mathbf{S} of subsets of \mathcal{U} , we are interested in finding the smallest subset \mathbf{T} of \mathbf{S} such that \mathbf{T} is a cover of \mathcal{U} , i.e. $\bigcup_{T \in \mathbf{T}} T = \mathcal{U}$. Because this problem cannot be approximated within a factor of $\Theta(\ln n)$ (see [2] and references therein), the following result proves Lemma 4.1:

Lemma D.1. *Given an α -approximation algorithm for the technology diffusion problem with constant number of threshold values $\theta \geq 2$, and constant graph diameter $r \geq 3$, we can obtain an $O(\alpha)$ -approximation algorithm for set cover problem. Moreover, the reduction holds even if the seedset in the technology diffusion problem is required to be connected.*

We remark that the main difficulty in constructing the reduction is that our utility function is non-local (i.e., a node may decide to activate because a remote node activated), while in typical NP complete problems, constraints are usually expressed in local form (i.e., they only depend on a small number of variables). To encode constraints of an NP complete problem into gadgets for a technology diffusion problem, we need to carefully insulate “influences” across different vertices so that node activations do not trigger an unplanned cascade. We do this using a padding argument. Roughly speaking, to protect against inadvertent activations of a vertex v , we replicate the vertices u that are *supposed* to activate v so that they block influences from other possibly-activated vertices connected to v .

of Lemma D.1. Let us consider an arbitrary a set cover instance $(\mathcal{U}, \mathbf{T})$, where $m = |\mathbf{T}|$ is the maximum number of sets in \mathbf{T} .

The reduction. We construct a technology diffusion problem as described below, and illustrated in Figure 10:

- The vertex set consists of the following types of vertices:
 1. The *set type*: for each $T \in \mathbf{T}$, we shall construct a vertex u_T in the technology network.
 2. The *element type*: for each $e \in \mathcal{U}$, we shall construct $m + 1$ vertices $u_{e,1}, u_{e,2}, \dots, u_{e,m+1}$.
- The edge set consists of the following edges:
 1. For each $T \in \mathbf{T}$ and $e \in T$, we add the edges $\{u_T, u_{e,1}\}, \{u_T, u_{e,2}\}, \dots, \{u_T, u_{e,m+1}\}$.
 2. The set type vertices are connected as a clique. (For each $T \neq T' \in \mathbf{T}$, we add the edge $\{u_T, u_{T'}\}$).
- The thresholds $\theta(\cdot)$ are set as follows,
 1. For any $e \in \mathcal{U}$ and $i \leq m + 1$, we set $\theta(u_{e,i}) = 2$.
 2. For every $T \in \mathbf{T}$, we set $\theta(u_T) = (m + 1)n + 1$.

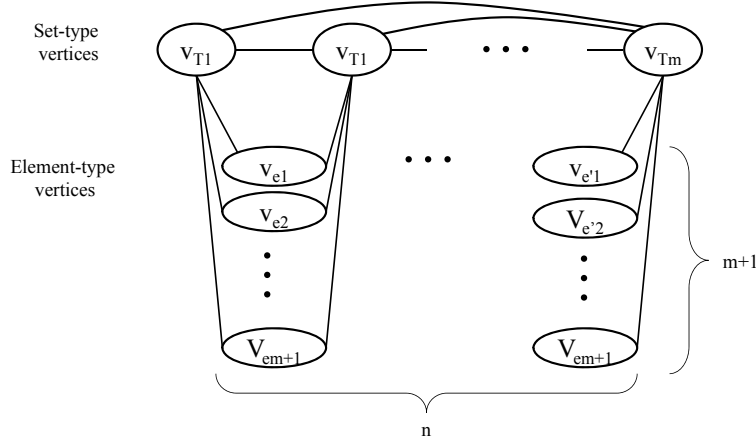


Figure 10: Reduction.

Properties of the reduction. Notice that our technology diffusion problem has only two types of threshold values. Furthermore, the diameter of the graph we form is exactly 3 hops (in terms of edges); the maximum distance in this graph is from one $u_{e,i}$ node to another. Finally, we show below that the seedset must consist of set-type vertices. Since these vertices form a clique, it follows that the seedset must be connected.

Correctness. To conclude that the size of the optimal seed set is the same as the size of the optimal cover (which also means that our reduction is approximation-preserving), we establish the following:

Item 1. For any feasible cover \mathbf{S} in the set cover problem, the corresponding seed set $\{u_S : S \in \mathbf{S}\}$ is a feasible solution for the technology diffusion problem.

Item 2. Any feasible seedset in the technology diffusion problem *that only consists of set-type vertices* corresponds with a feasible cover in the set cover problem.

Item 3. Given a feasible seedset that consists of *element type* vertices, there is an feasible seedset of equal or smaller size that consists only of *set type* vertices. Since the set type vertices form a clique, we have that the optimal solution for the technology diffusion problem is also a connected one.

Item 1. To show the first item, we simply walk through the activation process: When \mathcal{S} is a cover, let the seedset be u_{T_i} for all $T_i \in \mathcal{S}$. Notice that this seedset is *connected*. Upon activating the seedset, the vertices $u_{e,i}$ for all $e \in \mathcal{U}$ and $i \leq m+1$ are activated on because they are connected to at least one active seed. Now, there are $(m+1)n$ active nodes, so the rest of the *set type* vertices are activated.

Item 2. To show the second item, we consider an arbitrary seedset that only consists of the *set type* vertices: $U = \{u_{T_1}, u_{T_2}, \dots, u_{T_k}\}$, where $T_1, \dots, T_k \in \mathbf{T}$. We shall show that if T_1, \dots, T_k is not a cover, then the seed set cannot be feasible (*i.e.*, some nodes will remain inactive in the technology diffusion problem).

Let $e \in \mathcal{U} / (\cup_{j \leq k} T_j)$ be an element that is not covered by the sets in $\{T_1, \dots, T_k\}$. Let us consider the vertices $u_{e,1}, u_{e,2}, \dots, u_{e,m+1}$, and vertex u_T for each $T \notin \{T_1, \dots, T_k\}$ in the technology diffusion problem. We claim that none of these vertices will be activated with seedset U . Suppose, for the sake of contradiction, that one or more of these vertices are activated, and consider the first activated vertex among them. There are three cases:

Case 1. u_T ($T \notin \mathbf{T}$) is activated first. This is impossible: when $u_{e,i}$ ($i \leq m+1$) are not activated, the number of activated nodes is at most $(n-1)(m+1) + m < (m+1)n$.

Case 2. $u_{e,i}$ ($i \leq m+1$) is activated first. This is impossible because $u_{e,i}$ is only connected with u_T , where $T \notin \{T_1, \dots, T_k\}$ and none these *set type* vertices are activated.

Item 3. Finally, we move onto the third item. Let us consider a feasible seedset F that does not consist of only *set type* vertices. We show that we can easily remove the *element type* vertices in F : let $u_{e,i}$ be an arbitrary vertex in F . Then we can remove $u_{e,i}$ from F and add an u_T to F such that $e \in T$. This does not increase the cardinality of F . Furthermore, $u_{e,i}$ would still be activated, which implies that the updated F is still be a feasible seed set. \square

E Connectivity implies dependence on graph diameter r

We now prove Lemma 4.2, which shows that any algorithm that considers only connected seedsets, suffers a factor of r loss in the approximation rate:

of Lemma 4.2. Let $r > 0$ be an arbitrary integer. Let us define a line graph G_r as follows (Figure 11):

- The vertex set is $\{v_1, \dots, v_{2r+1}\}$.
- The edge set is $\{\{v_i, v_{i+1}\} : 1 \leq i < 2r+1\}$.

The threshold function shall be defined as follows,

- $\theta(v_1) = \theta(v_{2r+1}) = 2$ and $\theta(v_{r+1}) = 2r+1$.
- For $1 < i \leq r$, $\theta(v_i) = i$.
- For $r+2 \leq i < 2r+1$, $\theta(v_i) = 2r+2-i$.

It is straightforward to see that the diameter of the graph is $2r = \Theta(r)$. It remains to verify that the optimal connected solution is $\Theta(r)$ times larger than the optimal solution.

It's easy to see that $\{v_1, v_{2r+1}\}$ is a feasible seedset and therefore, the size of the optimal seed set is $O(1)$. We next show that any feasible connected set has size $\Omega(r)$.

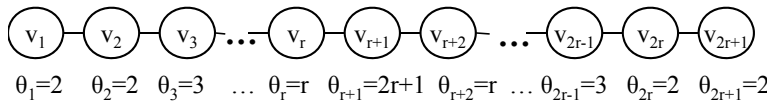


Figure 11: An instance of the technology diffusion problem for the proof of Lemma 4.2.

Since the seedset must be connected, wlog we can assume that the seedset is $\{v_i, v_{i+1}, \dots, v_j\}$ and by symmetry $i \leq r + 1$. When $j < r + 1$, node v_{r+1} will never activate (because v_{r+1} has threshold $2r + 1$, it only activates when all other nodes are active, but in the case all r nodes to the right of v_{r+1} are inactive). It follows that a feasible seedset requires $j \geq r + 1$.

When $i = 1$, the size of the seedset is $\Theta(r)$ and the lemma follows. So, need only consider the case where $i > 1$: symmetry allows us to assume wlog that $r + 1 - i \geq j - (r + 1)$ i.e., $\theta(v_{j+1}) \geq \theta(v_{i-1})$. Therefore, since we have $j - i + 1$ nodes in the seedset, a necessary condition for this seedset to be feasible is thus $j - i + 1 \geq i - 2$. Using the fact that $j \geq r + 1$, we get $i \leq r/2 + 2$ and $j - i = \Omega(r)$, which completes our proof. \square

One drawback of this construction is that $\ell = \Theta(n)$. We may modify $\theta(\cdot)$ so that $\ell = \tilde{O}(1)$ (thus ensuring that our lower bound depends on graph diameter r , rather than the number of thresholds ℓ):

- When $i \leq n$, set $\theta(u_i) = \max\{2^{\lfloor \log_2 i \rfloor}, 2\}$,
- when $i = n + 1$, set $\theta(u_i) = 2n + 1$, and
- When $i > n$, set $\theta(u_i) = \max\{2^{\lfloor \log_2(2n+2-i) \rfloor}, 2\}$.

One can use similar arguments to show that the size of the optimal seedset is $O(1)$ while the size of the optimal connected seedset is $\Theta(r)$.

F Our problem is neither submodular nor supermodular

We wondered about the relationship between the algorithmic properties of our model and the linear threshold model on social networks articulated in [23]. [7] showed that the problem of selecting an optimal seedset in the linear threshold mode in social networks cannot be approximated within a factor of $O(2^{\log^{1-\epsilon}|V|})$ when the thresholds are deterministic and known to the algorithm. [23] got around this lower bound by assuming that nodes' thresholds are chosen uniformly at random *after* the seedset is selected, and designing an algorithm that chooses the optimal seedset *in expectation*. Their $(1 - 1/e - \epsilon)$ -approximation algorithm relies on the submodularity of the *influence function*, i.e., the function $f(S)$ which gives the expected number of nodes that activate given that nodes in S are active.

In this section, we shall show that algorithmic results for submodular and/or supermodular optimization do *not* directly apply to our problem, even if we restrict ourselves to (a) graphs of constant diameter, (b) diffusion problems with a small number of fixed thresholds, or if (c) we choose the thresholds are uniformly at random as in [23]. Moreover, we see neither diminishing, nor increasing marginal returns even if we restrict ourselves to (d) connected seedsets.

F.1 Fixed threshold case

In this section, we construct two families of technology diffusion instances per the model in Definition H.8. Each family will be on a graph of diameter at most 4, and require at most 2 different threshold values, and each will consider connected seedsets. The first family will *fail* to exhibit the submodularity property while the second will *fail* to exhibit supermodularity.

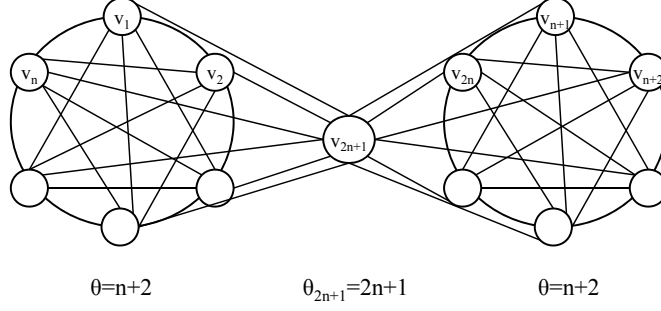


Figure 12: An instance of the technology diffusion problem.

Let $\{G, \theta\}$ be an arbitrary technology diffusion problem. We shall write $f_{G, \theta}(S)$ be the total number of nodes that are eventually activate after seedset S activates. When G and θ are clear from the context, we simply refer to $f_{G, \theta}(S)$ as $f(S)$.

F.1.1 The influence function is not submodular.

Let n be a sufficiently large integer such that the number of vertices in the graph is $2n+1$. This family of technology diffusion problems (which again is implicitly parameterized by n) is shown in Figure 12 and defined as follows:

- The vertex set is $\{v_1, v_2, \dots, v_{2n+1}\}$.
- The edge set is constructed as follows,
 - The subsets $\{v_1, \dots, v_n\}$ and $\{v_{n+1}, \dots, v_{2n}\}$ form two cliques.
 - Vertex v_{2n+1} is connected to all other vertices in the graph, *i.e.*, edges $\{v_1, v_{2n+1}\}, \dots, \{v_{2n}, v_{2n+1}\}$.
- The threshold function is
 - for $i \leq 2n$, $\theta(v_i) = n + 2$.
 - $\theta(v_{2n+1}) = 2n + 1$.

To show this problem is non-submodular, we shall find two disjoint sets S_1 and S_2 such that

$$f(S_1) + f(S_2) < f(S_1 \cup S_2) \quad (6)$$

We chose $S_1 = \{v_1, \dots, v_n\}$ and $S_2 = \{v_{2n+1}\}$. Note that S_1 and S_2 are connected, and that $f(S_1) = n$, $f(S_2) = 1$, while $f(S_1 \cup S_2) = 2n + 1$ so that (6) holds. \square

F.1.2 The influence function is not supermodular.

Let n be a sufficiently large integer that represents the number of vertices in the graph. Our family of technology diffusion problems G, θ (implicitly parameterized by n) shown in Figure 13 and defined as follows:

- The vertex set is $\{v_1, \dots, v_n\}$.

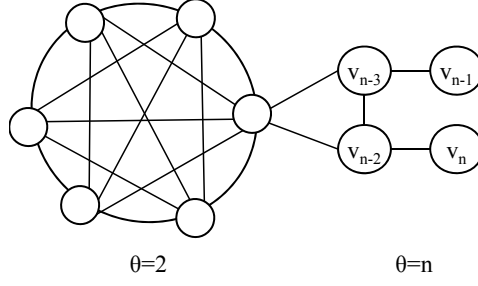


Figure 13: Another instance of the technology diffusion problem.

- The edge set is defined as follows:
 - For any $i < j \leq n - 4$, $\{v_i, v_j\}$ is in the edge set, *i.e.*, the subgraph induced by $\{v_1, \dots, v_{n-4}\}$ is a complete graph.
 - The remaining edges are $\{v_1, v_{n-3}\}$, $\{v_1, v_{n-2}\}$, $\{v_{n-3}, v_{n-1}\}$, $\{v_{n-2}, v_n\}$, and $\{v_{n-3}, v_{n-2}\}$.
- The threshold function is
 - For $i \leq n - 4$, $\theta(v_i) = 2$.
 - For $i > n - 4$, $\theta(v_i) = n$.

To show this problem is not supermodular, we choose two disjoint sets S_1 and S_2 such that

$$f(S_1) + f(S_2) > f(S_1 \cup S_2) \quad (7)$$

We choose $S_1 = \{v_{n-3}\}$ and $S_2 = \{v_{n-2}\}$. Note that S_1 and S_2 are connected, and $f(S_1) = f(S_2) = n - 3$, while $f(S_1 \cup S_2) = n - 2$ so that (7) indeed holds. \square

F.2 Randomized threshold case

We now consider a modified version of our problem, where, as in [23], we assume nodes thresholds are chosen uniformly at random:

Definition F.1 (Randomized technology diffusion optimization problem.). *The randomized technology diffusion model is identical to the model defined in Definition 2.1, with the exception that nodes choose their thresholds uniformly and independently at random from the set $\{2, 3, \dots, n\}$. Thus, the randomized technology diffusion optimization problem is to find the smallest feasible seedset S in expectation over node's choice of thresholds, when G is given as input.*

We follow [23] and let the influence function $f_G(S)$ be the *expected* number of vertices that are eventually activated, *i.e.*, $f_G(S) = \mathbb{E}_\theta[f_{G,\theta}(S)]$, where $f_{G,\theta}(S)$ is the number of activated vertices, and expectation is taken over the choice of thresholds. We present two families of problem instances: each family will be on a graph of diameter at most 4, and will consider connected seedsets. The first family will *fail* to exhibit submodularity of $f_G(S)$, while the second will *fail* to exhibit supermodularity.

F.2.1 The influence function is not submodular.

Let n be a sufficiently large integer such that the number of vertices in the network is $2n + 1$. Our family of G (parameterized by n) is defined as

- The vertex set is $\{v_1, v_2, \dots, v_{2n+1}\}$.
- The edge set is constructed as follows,
 - The subsets $\{v_1, \dots, v_n\}$ and $\{v_{n+1}, \dots, v_{2n}\}$ form two cliques.
 - The remaining edges are $\{v_{2n+1}, v_1\}$ and $\{v_{2n+1}, v_{2n}\}$.

Notice that this family of graphs is *almost* identical to the non-submodular example presented in the previous section, shown in Figure 12, except that now, the middle node v_{2n+1} is only connected to v_1 and v_{2n} . We shall find two disjoint set S_1 and S_2 such that

$$f_G(S_1) + f_G(S_2) \leq f_G(S_1 \cup S_2). \quad (8)$$

Our choice of S_1 and S_2 is $S_1 = \{v_1, \dots, v_n\}$ and $S_2 = \{v_{2n+1}\}$. We start with computing $f_G(S_1)$:

$$f_G(S_1) = \mathbb{E}[f_{G,\theta}(S_1) \mid \theta(v_{2n+1}) \leq n+1] \Pr[\theta(v_{2n+1}) \leq n+1] + \mathbb{E}[f_{G,\theta}(S_1) \mid \theta(v_{2n+1}) > n+1] \Pr[\theta(v_{2n+1}) > n+1] \quad (9)$$

Notice that

$$\begin{aligned} \mathbb{E}[f_{G,\theta}(S_1) \mid \theta(v_{2n+1}) \leq n+1] &= \mathbb{E}[f_{G,\theta}(S_1 \cup S_2)] = f_G(S_1 \cup S_2) \\ \mathbb{E}[f_{G,\theta}(S_1) \mid \theta(v_{2n+1}) > n+1] &= n \end{aligned} \quad (10)$$

Therefore, we may rewrite (9) as

$$f_G(S_1) = f_G(S_1 \cup S_2) \Pr[\theta(v_{2n+1}) \leq n+1] + n \Pr[\theta(v_{2n+1}) > n+1] = \frac{f_G(S_1 \cup S_2)}{2} + \frac{n}{2}. \quad (11)$$

We next move to compute $f_G(S_2)$. To understand how the influence of $S_2 = \{v_{2n+1}\}$ spreads, we condition on the thresholds of its neighbors: $\theta(v_1), \theta(v_{2n})$.

$$\begin{aligned} f_G(S_2) &\leq 1 \cdot \Pr[\theta(v_1) > 2 \cap \theta(v_{2n}) > 2] + (2n+1) \cdot \Pr[\theta(v_1) = 2 \cup \theta(v_{2n}) = 2] \\ &= 1(1 - \frac{1}{2n})(1 - \frac{1}{2n}) + (2n+1)(2\frac{1}{2n}(1 - \frac{1}{2n}) + \frac{1}{2n}\frac{1}{2n}) \\ &= 1 + 2n\frac{1}{2n}(2(1 - \frac{1}{2n}) + \frac{1}{2n}) \leq 3 \end{aligned} \quad (12)$$

Therefore, from (11) and (12) we have

$$f_G(S_1) + f_G(S_2) \leq 3 + \frac{1}{2}(f_G(S_1 \cup S_2) + n) \quad (13)$$

Recall that our goal is to show that $f_G(S_1) + f_G(S_2) < f_G(S_1 \cup S_2)$. Using (13), we now see that it suffices to prove that

$$f_G(S_1 \cup S_2) > n + 6$$

We prove this by conditioning on the event that $S_1 \cup S_2$ activates node v_{2n} :

$$\begin{aligned} f_G(S_1 \cup S_2) &= f_G(S_1 \cup S_2 \cup \{v_{2n}\}) \Pr[\theta(v_{2n}) \geq n+1] + (n+1) \Pr[\theta(v_{2n}) < n+1] \\ &\geq (n+2 + \frac{n-1}{2})\frac{n+1}{2n} + (n+1)\frac{n-1}{2n} \\ &= n+1 + \frac{n+1}{4} \end{aligned}$$

where the first inequality follows because the thresholds of half of the nonseed vertices $\{v_{n+1}, \dots, v_{2n-1}\}$ are $\leq n+1$ in expectation. Thus, we indeed have that S_1 and S_2 are connected and $f_G(S_1) + f_G(S_2) < f_G(S_1 \cup S_2)$ when n is sufficiently large. \square

F.2.2 The influence function is not supermodular.

Let n be a sufficiently large integer such that the number of vertices in the network is $2n + 1$. Our family of G (parameterized by n) is defined as follows,

- The vertex set is $\{v_1, v_2, \dots, v_{2n+1}\}$.
- The edge set is constructed as follows,
 - The subsets $\{v_1, \dots, v_n\}$ and $\{v_{n+1}, \dots, v_{2n}\}$ form two cliques.
 - Vertex v_{2n+1} is connected to all other nodes in the graph.
 - There is an additional edge $\{v_1, v_{2n}\}$.

Notice that this family of graphs is almost identical to the one shown in Figure 12, except for the addition of a single edge $\{v_1, v_{2n}\}$. We shall find two disjoint set S_1 and S_2 such that

$$f_G(S_1) + f_G(S_2) > f_G(S_1 \cup S_2). \quad (14)$$

Our choice of S_1 and S_2 is $S_1 = \{v_1, \dots, v_n\}$ and $S_2 = \{v_{n+1}, \dots, v_{2n}\}$. Notice that these sets are connected by the edge $\{v_1, v_{2n}\}$. By symmetry we have that $f(S_1) = f(S_2)$, so we start by computing $f_G(S_1)$. Let T be the number of active nodes in S_2 , and let A be the event that vertex v_{2n+1} is active.

$$\begin{aligned} \mathbb{E}[f_{G,\theta}(S_1)] &\geq n + (1 + \mathbb{E}[T|A, S_1 \text{ active}]) \Pr[A|S_1 \text{ active}] \\ &\geq n + (1 + n \cdot \frac{n+1}{2n}) \frac{n}{2n} \\ &= n + \frac{1}{2}(1 + \frac{n+1}{4}) \end{aligned} \quad (15)$$

where the second inequality follows because we used the trivial bound $\mathbb{E}[T|A, S_1 \text{ active}] \geq n \frac{n+1}{2n}$ where we ignore all cascading effects; we simply assume that each of the n nodes in S_2 is connected to an active component of size $n + 1$. On the other hand,

$$\mathbb{E}[f_{G,\theta}(S_1 \cup S_2)] = 2n + \Pr[A|S_1 \cup S_2 \text{ active}] = 2n + 1$$

Thus we indeed have $f_G(S_1) + f_G(S_2) \geq 2n + 1 + \frac{n+1}{4} > 2n + 1 = f_G(S_1 \cup S_2)$ for all n . \square

G Probability review

Theorem G.1 (Chenroff bounds). *Let X_1, \dots, X_n be independent Poisson trials such that $\Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[X]$. Then*

1. For $0 < \delta < 1$,

$$\Pr[|X - \mu| \geq \delta\mu] \leq 2 \exp(-\mu\delta^2/3).$$

2. for $R \geq 6\mu$,

$$\Pr[X \geq R] \leq 2^{-R}.$$

H Lemmas to prove Theorem 3.3

In the following, let \mathcal{H} be the hypergraph corresponding to an optimal solution σ to our relaxed LP in Figure 2, and let $S_0 \leftarrow \text{PRELIM-SEEDSET}(\mathcal{H})$ and $T_0 \leftarrow \text{GET-SEQ}(\mathcal{H}, S_0)$. Also, for each $k \in \{1, \dots, \ell\}$, let $S_k \leftarrow \text{UPDATE-SEEDSET}(\mathcal{H}, S_{k-1})$ and $T_k \leftarrow \text{GET-SEQ}(\mathcal{H}, S_k)$ for S_{k-1} that satisfies conditions (C.1)-(C.2).

H.1 Size of preliminary seedset S_0 .

We argue that, with high probability, PRELIM-SEEDSET gives us a preliminary seedset S_0 that is at most $O(r \ln n)$ times the one given by the optimal solution opt .

Lemma H.1. *Let $S_0 \leftarrow \text{PRELIM-SEEDSET}(\mathcal{H})$. We have*

$$\Pr[|S_0| \geq 24(1 + \epsilon)^2 r (\ln 2n) (2\text{opt})] = o(1).$$

Proof. We shall show that PRELIM-SEEDSET selects at most $24(1 + \epsilon)^2 \ln(2n)|\sigma|$ seeds with high probability, where by $|\sigma|$ we mean the value of the objective function of the linear program (which recall from Lemma 3.2 is of size at most (2opt)), so that $|\sigma| = \sum_{i \leq n} \sum_{t \leq \theta(u_i)} x_{i,t}$. The lemma follows from the fact that GLUE-SEEDS used inside PRELIM-SEEDSET expands the seed set at most a r factor, and $|\sigma| \leq (2\text{opt})$.

Now let Z_i be the indicator random variable that sets to 1 if u_i is selected as seed during *the second step* of PRELIM-SEEDSET (*i.e.*, before gluing). Then we have $\Pr[Z_i = 1] = \min\{1, 24 \ln(2n) \sum_{t \leq \theta(u_i)} x_{i,t}\}$. It follows that

$$\mathbb{E}[\sum_{i \leq n} Z_i] \leq 24(1 + \epsilon) \ln(2n) \sum_{i \leq n} \sum_{t \leq \theta(u_i)} x_{i,t} = 24(1 + \epsilon) \ln(2n) |\sigma|.$$

Since Z_i are chosen independently, we may apply a Chernoff bound (Theorem G.1) and get

$$\Pr[\sum_{i \leq n} Z_i \geq 24(1 + \epsilon)^2 \ln(2n) |\sigma|] \leq \exp\left(-\mathbb{E}[\sum_{i \leq n} Z_i] \epsilon^2 / 2\right) = o(1).$$

and so our lemma follows. \square

H.2 Feasibility of preliminary activation sequence T_0 .

Our lemma below addresses condition (P.1), and therefore the feasibility of T_0 . We show that, during the GET-SEQ procedure, there should be at least one flag such that $b_{i,t} = 1 \ \forall i$ with good probability.

Lemma H.2. *Let T_0 be obtained as described above. Consider an arbitrary $i \in [n]$. Let*

$$t(i) = \min\{t : \sum_{t' \leq t} x_{it'} \geq \frac{1}{12(1 + \epsilon)}\}$$

It follows GET-SEQ assigns the flags $b_{i,t}$ such that

$$\Pr \left[\exists i : \bigwedge_{t' \leq t(i)} (b_{i,t'} = 0) \right] \leq \frac{1}{2}.$$

The proof of Lemma H.2 relies on some new notation and technical lemmas. First, for each $i, t \in [n]$, let an arbitrary solution $\mathcal{F}_{i,t}$ for the (i, t) -flow problem be the *representative flow* for the (i, t) -flow problem. We also need the notion of *border nodes*:

Definition H.3 (Border nodes for the (i, t) -flow). *Consider the (i, t) -flow problem on the hypergraph \mathcal{H} and the corresponding $\mathcal{F}_{i,t}$. Let us decompose $\mathcal{F}_{i,t}$ into paths (in an arbitrary but consistent manner) $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_q$. Consider an arbitrary one of these pathes \mathcal{P}_j . Let X_{i_j, t_j} be the last vertex on \mathcal{P}_j such that X_{i_j, t_j} is to the left of the threshold line. We denote X_{i_j, t_j} as $\text{border}(\mathcal{P}_j)$. The border nodes of the (i, t) -flow problem is the set of vertices $\beta(i, t) = \{\text{border}(\mathcal{P}_1), \dots, \text{border}(\mathcal{P}_q)\}$.*

We refer to Figure 8 to illustrate an example. Consider the representative flow $\mathcal{F}_{E,6}$ for the $(E, 6)$ -flow problem. Suppose the flow decomposes into three pathes, $\mathcal{P}_1 = X_{A,1} \rightarrow X_{C,3} \rightarrow X_{F,4} \rightarrow X_{C,5} \rightarrow X_{E,6}$, $\mathcal{P}_2 = X_{A,1} \rightarrow X_{C,3} \rightarrow X_{F,4} \rightarrow X_{E,6}$, and $\mathcal{P}_3 = X_{A,1} \rightarrow X_{B,5} \rightarrow X_{E,6}$. We have $\text{border}(\mathcal{P}_1) = X_{F,4}$ because $X_{F,4}$ is the last time the path stays to the left of the threshold. Also, $\text{border}(\mathcal{P}_2) = X_{F,4}$, and $\text{border}(\mathcal{P}_3) = X_{B,5}$. Therefore, for this example, $\beta(E, 6) = \{X_{B,5}, X_{F,4}\}$.

Lemma H.4. *Consider an arbitrary (i, t) -flow problem for the graph \mathcal{H} and its corresponding set of border nodes $\beta(i, t)$. We have that the sum of the capacity of all the border nodes in $\beta(i, t)$ is at least as large as the sum of demands of the sinks $X_{i,t'}$ for $\theta(u_i) \leq t' \leq t$, i.e.,*

$$\sum_{X_{j,t'} \in \beta(i,t)} x_{j,t'} \geq \sum_{\theta(u_i) \leq t' \leq t} x_{i,t'}.$$

of Lemma H.4. Consider the representative flow $\mathcal{F}_{i,t}$ for the (i, t) problem. Let $|\mathcal{F}_{i,t}|$ be the corresponding volume of the flow and let $\mathcal{F}_{i,t}(X)$ be the volume of the flow on the node X . We have

$$\sum_{X \in \beta(i,t)} \mathcal{F}_{i,t}(X) = |\mathcal{F}_{i,t}| = \sum_{\theta(u_i) \leq t' \leq t} x_{i,t'}.$$

On the other hand, we also require the capacity of any node X is no less than the actual flow $|\mathcal{F}_{i,t}(X)|$. Therefore,

$$\sum_{X_{j,t'} \in \beta(i,t)} x_{j,t'} \geq \sum_{X \in \beta(i,t)} \mathcal{F}_{i,t}(X).$$

The lemma therefore follows. □

We are now ready to prove Lemma H.2.

of Lemma H.2. Let $\beta(i, t(i))$ be the border nodes of the flow $\mathcal{F}_{i,t(i)}$, let

$$\omega_i = \sum_{t < \theta(u_i)} x_{i,t} \tag{16}$$

be the mass of node u_i to the left of the threshold line, and let

$$p_i = \min\{(24(1 + \epsilon) \ln(2n))\omega_i, 1\} \quad (17)$$

be the probability with which PRELIM-SEEDSET fixes node u_i as a seed.

A node u_i turns on before timestep $t(i)$ in activation function T_0 , *i.e.*, $\bigvee_{t' \leq t(i)} (b_{i,t'} = 1)$, if either:

1. u_i is selected as a seed; this happens with probability p_i .
2. One of the border nodes $X_{j,t'} \in \beta(i, t(i))$ corresponds to a seed node u_j ; this happens with probability p_j .

Notice that the above two events are not necessarily independent, because $X_{i,t'}$ could be a border node in some $(i, t(i))$ -flow problem.⁴ Next, we have

$$\begin{aligned} \Pr \left[\bigwedge_{t' \leq t(i)} (b_{i,t'} = 1) \right] &\leq \min\{1 - p_i, \prod_{\substack{j: \exists t' \\ X_{j,t'} \in \beta(i,t)}} (1 - p_j)\} \\ &= \min \left\{ 1 - \min\{(24(1 + \epsilon) \ln(2n))\omega_i, 1\}, \prod_{\substack{j: \exists t' \\ X_{j,t'} \in \beta(i,t)}} (1 - \min\{(24(1 + \epsilon) \ln(2n))\omega_j, 1\}) \right\} \end{aligned} \quad (18)$$

We use Lemma H.4 and some algebra to bound the quantity on the right hand side of (18). Roughly, we use the idea discussed in Section 3.1: if the total flow at node u at time t is $f_1 + f_2$, then the probability that the technology is diffused to u via either of these two flows is $1 - (1 - f_1)(1 - f_2) \approx f_1 + f_2$, so that the total flow can be used to determine node u 's activation probability. We start by noticing that

$$\sum_{\substack{j: \exists t' \\ X_{j,t'} \in \beta(i,t)}} \omega_j = \sum_{\substack{j: \exists t' \\ X_{j,t'} \in \beta(i,t)}} \sum_{t < \theta(u_j)} x_{j,t} \geq \sum_{\substack{j, t': \\ X_{j,t'} \in \beta(i,t)}} x_{j,t'} \geq \sum_{\theta(u_i) \leq t' \leq t(i)} x_{i,t'} \quad (19)$$

(where the first equality holds by equation (16), and the last inequality holds because of Lemma H.4.) Therefore, we have

$$2 \max \left\{ \omega_i, \sum_{\substack{j: \exists t' \\ X_{j,t'} \in \beta(i,t)}} \omega_j \right\} \geq \omega_i + \sum_{\substack{j: \exists t' \\ X_{j,t'} \in \beta(i,t)}} \omega_j \geq \sum_{t' \leq \theta(u_i)} x_{i,t'} + \sum_{\theta(u_i) \leq t' \leq t(i)} x_{i,t'} \geq \sum_{t' \leq t(i)} x_{i,t'} \geq \frac{1}{12(1 + \epsilon)} \quad (20)$$

(where the first inequality follows from algebra, the second from equations (16) and (19), the third from algebra, and the final by definition of timestep $t(i)$ in the statement of this lemma.) Our next step involves an arithmetic lemma, as follows:

⁴While at first glance this seems to suggest the type of problematic recirculation of flow leading to the integrality gap we discussed in Appendix A, it is not actually a problem for us, since the flow through such an $X_{i,t'}$ cannot be amplified due to the flow constraints.

Lemma H.5. Let ϵ be a suitably small constant. Let x_1, \dots, x_n be numbers between $[0, 1]$ such that $\sum_{i \leq n} x_i = s$, where $s \geq \frac{1}{24(1+\epsilon)}$. Let $\lambda = 24(1 + \epsilon) \ln(2n)$. It follows that

$$\prod_{i \leq n} (1 - \min\{\lambda x_i, 1\}) \leq \frac{1}{2n}.$$

Equation (20) allows us to apply the arithmetic lemma to the right side of (18) as follows:

$$\min \left\{ 1 - \min\{(24(1 + \epsilon) \ln(2n))\omega_i, 1\}, \prod_{\substack{j: \exists t' \\ X_{j,t'} \in \beta(i,t)}} (1 - \min\{(24(1 + \epsilon) \ln(2n))\omega_j, 1\}) \right\} \leq \frac{1}{2n}$$

and the lemma follows from a union bound across all nodes. \square

of Arithmetic Lemma H.5. Let us consider two cases over the values of x_i . In the first case, there exists some x_i such that $\lambda x_i \geq 1$. For this case, we have

$$\prod_{i \leq n} (1 - \min\{\lambda x_i, 1\}) = 0 \leq \frac{1}{2n}.$$

In the second case, where all x_i are less than $1/\lambda$, the quantity $\prod_{i \leq n} (1 - \min\{\lambda x_i, 1\}) = \prod_{i \leq n} (1 - \lambda x_i)$ is maximized when $x_1 = x_2 = \dots = x_n = \frac{s}{n}$. In other words,

$$\begin{aligned} \prod_{i \leq n} (1 - \lambda x_i) &\leq \left(1 - \frac{\lambda s}{n}\right)^n \\ &= \left(1 - \frac{\lambda s}{n}\right)^{\frac{n}{\lambda s} \lambda s} \\ &\leq \exp(-\lambda s) \\ &\leq \exp\left(-\frac{\lambda}{24(1 + \epsilon)}\right) = \exp(-\ln(2n)) = \frac{1}{2n}. \end{aligned}$$

\square

H.3 Connectivity Lemma for the activation sequence.

Next we prove the Connectivity Lemma, that shows that if GET-SEQ takes in a connected seedset, it returns a connected activation sequence:

of Connectivity Lemma 3.5. We inductively prove that $\bigcup_{t' \leq t} T^{-1}(t')$ is connected for $1 \leq t \leq n$. For the base case, observe that GET-SEQ is such that $T^{-1}(1) = S$ is the set of seeds, so $T^{-1}(1)$ is connected. For any non-seed node u_i such that $T(u_i) > 1$, GET-SEQ is such that the corresponding hypergraph node $X_{i,T(u_i)}$ to the right of the threshold line is “activated”. Further, there is a path in \mathcal{H} from $X_{i,T(u_i)}$ to some “activated” hypergraph node $X_{j,t}$ to the left of the threshold line, where $u_j \in S$ is a seed and $t < \theta(u_j)$. Each edge in \mathcal{H} corresponds to an edge in G , and all hypergraphs nodes along the path must be “activated” before $T(u_i)$. Thus, u_i is connected to the seedset S in the subgraph of G induced by $\{u_i\} \cup \left(\bigcup_{t < T(u_i)} T^{-1}(u_i)\right)$. Connectivity of T follows. \square

H.4 Partial consistency.

The following lemma shows that partial consistency, *i.e.*, condition (C.1), is almost immediate:

Lemma H.6. S_k and T_k are partially consistent up to $\theta_k - 1$ if

1. S_{k-1} and T_{k-1} are partially consistent up to time $\theta_{k-1} - 1$, and
2. T_k satisfies (C.2); *i.e.*, T_k has at least $\theta_j - 1$ nodes active by timestep $\theta_j - 1$ for any $j < k$.

Proof. First, the Connectivity Lemma 3.5 implies that T_k is connected. Therefore, to decide whether a node u_i is a seed with respect to T_k , we need count the number of active nodes (per T_k) prior to time $T_k(u_i)$ and compare it with u_i 's threshold $\theta(u_i)$. Thus, we need only prove that for any node u_i such that $\theta_{k-1} \leq T_k(u_i) \leq \theta_k - 1$, either (a) the number of active nodes prior to time $T_k(u_i)$ is at least $\theta(u_i) - 1$ or (b) u_i is a seed, *i.e.*, $u_i \in S_k$. We have two cases:

Case 1. $T_k(u_i) < \theta(u_i)$. By construction of GET-SEQ, it follows that $u_i \in S_k$ is a seed.

Case 2. $T_k(u_i) \geq \theta(u_i)$. Since we are only concerned with nodes such that $T_k(u_i) < \theta_k$, we have $\theta(u_i) \in \{\theta_1, \dots, \theta_{k-1}\}$. Since T_k satisfies (C.2) by assumption, there are at least $\theta_j - 1$ active nodes at time $\theta_j - 1$ in T_k . Consequently, the number of activated nodes by the time step $T_k(u_i) - 1$ is at least $\theta(u_i) - 1$, (*i.e.*, T_k encodes u_i as a non-seed). \square

H.5 Proof of Gap Size Lemma 3.8

of Gap Size Lemma 3.8. This proof follows almost completely from algebra. First, recall that our LP requires that $\sum_{i \leq n} x_{i,t} = 1$ for all t . Therefore, simple algebra gives that $\theta_k - 1 = \sum_{t \leq \theta_k - 1} \sum_{i \leq n} x_{i,t}$. The same simple algebra allows us to write

$$|\{u_i : T_{k-1}(u_i) \leq \theta_k - 1\}| = \sum_{u_i : T_{k-1}(u_i) < \theta_k} 1 = \sum_{u_i : T_{k-1}(u_i) < \theta_k} \sum_{t \leq n} x_{i,t}.$$

Substituting in these expressions into (5), we have

$$\rho = \theta_k - 1 - |\{u_i : T_{k-1}(u_i) \leq \theta_k - 1\}| = \left(\sum_{i \leq n} \sum_{t \leq \theta_k - 1} x_{i,t} \right) - \left(\sum_{u_i : T_{k-1}(u_i) < \theta_k} \sum_{t \leq n} x_{i,t} \right)$$

Next, since $\theta_k - 1 < n$, we modify the second summation in the second summand to obtain

$$\begin{aligned} &\leq \left(\sum_{i \leq n} \sum_{t \leq \theta_k - 1} x_{i,t} \right) - \left(\sum_{u_i : T_{k-1}(u_i) < \theta_k} \sum_{t \leq \theta_k - 1} x_{i,t} \right) \\ &= \sum_{t \leq \theta_k - 1} \left(\left(\sum_{i \leq n} x_{i,t} \right) - \left(\sum_{u_i : T_{k-1}(u_i) < \theta_k} x_{i,t} \right) \right) \quad (\text{moving } t \text{ index ahead}). \end{aligned}$$

Finally, consider the term inside the first sum. Its first summand is over all vertices u_i $i = 1 \dots n$, while its second summand over all vertices u_i such that $T_{k-1}(u_i) < \theta_k$. Thus the difference between these summands is over all vertices u_i such that $T_{k-1}(u_i) \geq \theta_k$ so finally we have

$$= \sum_{t \leq \theta_k - 1} \left(\sum_{u_i : T_{k-1}(u_i) \geq \theta_k} x_{i,t} \right) = \gamma$$

□

H.6 Proof of Lemma 3.9

Lemma 3.9 is straightforward given the following Lemma H.7, as we shall show in Section H.6.2. In this section we focus our main task of proving Lemma H.7. Recall that the “gap” is given by

$$\gamma = \sum_{u_i: T_{k-1}(u_i) \geq \theta_k} \sum_{t < \theta_k} x_{i,t}$$

The following lemma handles with each “row” of the gap (*i.e.*, $u_i : T_{k-1}(u_i) \geq \theta_k$) separately:

Lemma H.7. *Let u_i be an arbitrary node such that $T_{k-1}(u_i) \geq \theta_k$, so that u_i is a candidate for moving forward in activation sequence T_k (relative to activation sequence T_{k-1}). It follows that*

$$\Pr[T_k(u_i) < \theta_k] \geq (1 + \epsilon) \sum_{t < \theta_k} x_{i,t}.$$

Before we begin the proof, we need a few definitions, related to the definitions we introduced in Appendix H.2:

Definition H.8 (Border flow). *Consider the (i, t) -flow problem and the corresponding representative flow $\mathcal{F}_{i,t}$. Let $\mathcal{P}_1, \dots, \mathcal{P}_q$ be the decomposition of $\mathcal{F}_{i,t}$ so that $\beta(i, t) = \{\text{border}(\mathcal{P}_1), \dots, \text{border}(\mathcal{P}_q)\}$. Fix an arbitrary $X \in \beta(i, t)$, define the flow across the border with respect to $\mathcal{F}_{i,t}$ as*

$$f_{i,t}(X) = \sum_{j: \text{border}(\mathcal{P}_j) = X} |\mathcal{P}_j|.$$

We shall refer $f_{i,t}(\cdot)$ as the border flow function with respect to the (i, t) -flow problem.

Notice that it is possible that border flow $f_{i,t}(X)$ does not equal the representative flow $\mathcal{F}_{i,t}$ that passes through X (*i.e.* $f_{i,t}(X) < \mathcal{F}_{i,t}(X)$) because, *e.g.*, there could be two paths from $\mathcal{P}_1, \dots, \mathcal{P}_q$ that passes through X where X is the border of one path and is not the border of the other one.

We refer back to Figure 8 for an example. Consider the $(B, 6)$ -flow problem and let $\mathcal{F}_{B,6}$ consist of two paths: $\mathcal{P}_1 = X_{A,1} \rightarrow X_{B,5}$ and $\mathcal{P}_2 = X_{A,1} \rightarrow X_{C,3} \rightarrow X_{F,4} \rightarrow X_{E,5} \rightarrow X_{B,6}$. Notice that $\beta(B, 6) = \{X_{A,1}, X_{F,4}\}$. We have $\mathcal{F}_{B,6}(X_{A,1}) = |\mathcal{F}_{B,6}|$ while $f_{B,6}(X_{A,1})$ only consists of the volume for flow along the path \mathcal{P}_1 . *i.e.*, $\mathcal{F}_{B,6}(X_{A,1}) \neq f_{B,6}(X_{A,1})$.

Our analysis for Lemma H.7 utilizes the following fact.

Fact H.1. *Consider the (i, t) -flow problem on the graph \mathcal{H} and the corresponding border flow function $f_{i,t}(\cdot)$. We have*

$$\sum_{X \in \beta(i, t)} f_{i,t}(X) = \sum_{\theta(u_i) \leq t} x_{i,t}. \quad (21)$$

This fact is intuitively straightforward, because all the “border flow” shall eventually move to the sinks, though the actual formalization is fairly tedious, so we present it after the proof of Lemma H.7:

of Lemma H.7. We concern ourselves with u_i that activates after timestep θ_k in activation function T_{k-1} . Let's consider the $(i, \theta_k - 1)$ -flow problem, and the corresponding border nodes $\beta(i, \theta_k - 1)$ as defined in Definition H.3. Since u turns on after θ_k in T_{k-1} , it must follow that none of the border nodes are activated in T_{k-1} ; otherwise, GET-SEQ would have activated u_i by timestep $\theta_k - 1$ in T_{k-1} .

Now let's consider T_k . A sufficient condition for u_i to be activated in T_k by timestep $\theta_k - 1$ is either (a) u_i is a seed, or (b) at least one node u_j corresponding to a border nodes in $\beta(i, \theta_k - 1)$ is selected as a seed. (Since by definition, border nodes $X_{j,t}$ always have $t < \theta(u_j)$, GET-SEQ is such that the only way a border node $X_{j,t}$ can be activated is if it corresponds to a seed u_j .)

Now node u_i will be activated before time θ_k in T_k if either (a) the border nodes $X_{j,t} \in \beta(i, \theta_k - 1)$ that were not "active" in T_{k-1} become active in T_k (this occurs with probability $4(1+\epsilon)\omega_j$, where ω_j is as in equation (16)) since $T_k \leftarrow \text{GET-SEQ}(\mathcal{H}, S_{k-1})$ and $S_k \leftarrow \text{UPDATE-SEEDSET}(\mathcal{H}, S_{k-1})$, or (b) u_i is itself is a seed in S_k (this occurs with probability $4(1+\epsilon)\omega_i$). Notice that events (a) and (b) could be correlated, so we have that

$$\Pr[T_k(u_i) \geq \theta_k] \leq \min \left\{ (1 - 4(1+\epsilon)\omega_i), \prod_{u_j \in \beta(i, \theta_k - 1)} (1 - 4(1+\epsilon)\omega_j) \right\} \quad (22)$$

Given equation (22), our lemma will follow from the following claim:

Claim H.1.

$$\min \left\{ (1 - 4(1+\epsilon)\omega_i), \prod_{u_j \in \beta(i, \theta_k - 1)} (1 - 4(1+\epsilon)\omega_j) \right\} \leq 1 - (1+\epsilon) \sum_{t < \theta_k} x_{i,t}. \quad (23)$$

□

Roughly speaking, the idea in Claim H.1 is to use the first order approximation to give a bound on the product term $\prod_{u_j \in \beta(i, \theta_k - 1)} (1 - 4(1+\epsilon)\omega_j)$. By only considering linear terms in this quantity, we get $\prod_{u_j \in \beta(i, \theta_k - 1)} (1 - 4(1+\epsilon)\omega_j) \approx 1 - O(\sum_{u_j \in \beta(i, \theta_k - 1)} \omega_j)$. Together with the inequality established in Fact H.1, we can rewrite $O(\sum_{u_j \in \beta(i, \theta_k - 1)} \omega_j) = O(\sum_{\theta(u_i) \leq t' \leq t} x_{i,t'})$, which allows us to conclude Claim H.1. We now formalize this idea step by step:

of Claim H.1. We start with analyzing the right term in product in inequality (23). Recall that $\mathcal{F}_{i, \theta_k - 1}$ is the representative flow for the $(i, \theta_k - 1)$ -flow problem. $\mathcal{F}_{i, \theta_k - 1}(X_{j,t'})$ is the corresponding flow that passes through the node $X_{j,t'}$. For each flow

$$\mathcal{F}_{i, \theta_k - 1}(X_{j,t'}) \leq x_{j,t'}$$

since $x_{j,t'}$ represents the capacity of the node $X_{j,t'}$ per the flow constraints. Now we consider the

terms inside the product on the left in equation (23):

$$\begin{aligned}
1 - 4(1 + \epsilon)\omega_j &= 1 - 4(1 + \epsilon) \left(\sum_{t < \theta(u_j)} x_{j,t} \right) \quad (\text{Definition of } \omega_j \text{ in equation (16)}) \\
&\leq 1 - 4(1 + \epsilon) \left(\sum_{t < \theta(u_j)} \mathcal{F}_{i, \theta_k - 1}(X_{j,t}) \right) \quad (\text{Flow is bounded by capacity}) \\
&\leq 1 - 4(1 + \epsilon) \left(\sum_{\substack{t < \theta(u_j) \wedge \\ X_{j,t} \in \beta(i, \theta_k - 1)}} \mathcal{F}_{i, \theta_k - 1}(X_{j,t}) \right) \quad (\text{algebra}) \\
&\leq 1 - 4(1 + \epsilon) \left(\sum_{\substack{t < \theta(u_j) \wedge \\ X_{j,t} \in \beta(i, \theta_k - 1)}} f_{i, \theta_k - 1}(X_{j,t}) \right) \quad (\text{Construction of } f_{i,t}(\cdot) \text{ in Definition H.8})
\end{aligned} \tag{24}$$

and notice that $X_{j,t}$ could be in $\beta(i, \theta_k - 1)$ only if $X_{j,t}$ is to the left of the threshold line, i.e. $t < \theta(u_j)$. Therefore, $X_{j,t} \in \beta(i, \theta_k - 1)$ implies $t < \theta(u_j)$ so we can write

$$= 1 - 4(1 + \epsilon) \sum_{t: X_{j,t} \in \beta(i, \theta_k - 1)} f_{i, \theta_k - 1}(X_{j,t}) \tag{25}$$

Next, let us analyze the term $(1 - 2(1 + \epsilon)\omega_i)$ in (23). We have

$$\begin{aligned}
1 - 4(1 + \epsilon)\omega_i &= 1 - 4(1 + \epsilon) \left(\sum_{t < \theta(u_i)} x_{i,t} \right) \quad (\text{Definition of } \omega_i \text{ in (16)}) \\
&\leq 1 - 4(1 + \epsilon) \left(\sum_{t < \min\{\theta_k - 1, \theta(u_i)\}} x_{i,t} \right) \quad (\text{Algebra})
\end{aligned} \tag{26}$$

We can substitute (26) and (25) back into our original equation (23), to yield a giant product. We won't write down this messy product yet. Instead, we show how to clean it up using approximation of its lower order terms. Specifically, we shall use linear terms to approximate the giant product as follows:

Lemma H.9 (Another Arithmetic Lemma). *Let x_1, x_2, \dots, x_n be real values such that $\sum_{i \leq n} x_i \leq \frac{1}{2}$. We have*

$$\prod_{i \leq n} (1 - x_i) \leq 1 - \left(\sum_{i \leq n} x_i \right) \left(1 - \frac{\sum_{i \leq n} x_i}{1 - \sum_{i \leq n} x_i} \right).$$

Specifically, when $\sum_{i \leq n} x_i \leq \frac{1}{3}$, we have

$$\prod_{i \leq n} (1 - x_i) \leq 1 - \frac{1}{2} \left(\sum_{i \leq n} x_i \right).$$

We present a proof of this arithmetic lemma after we complete the current proof. To use the arithmetic lemma to clean up our product, we need to show that condition specified in the lemma holds for our setting:

Claim H.2.

$$4(1 + \epsilon) \sum_{t < \min\{\theta_k - 1, \theta(u_i)\}} x_{i,t} + 4(1 + \epsilon) \sum_{\substack{t: \\ X_{j,t} \in \beta(i, \theta_k - 1)}} f_{i, \theta_k - 1}(X_{j,t}) \leq \frac{1}{3}. \quad (27)$$

Proof. Starting with the left side of (27), we can write

$$\begin{aligned} \sum_{t < \min\{\theta_k - 1, \theta(u_i)\}} x_{i,t} + \sum_{\substack{t: \\ X_{j,t} \in \beta(i, \theta_k - 1)}} f_{i, \theta_k - 1}(X_{j,t}) &= \sum_{t < \min\{\theta_k - 1, \theta(u_i)\}} x_{i,t} + \sum_{\theta(u_i) \leq t \leq \theta_k - 1} x_{i,t} \quad (\text{Using Fact H.1}) \\ &= \sum_{t < \theta(u_i)} x_{i,t} \quad (\text{Algebra}) \\ &< \frac{1}{12(1 + \epsilon)} \end{aligned} \quad (28)$$

$$< \frac{1}{12(1 + \epsilon)} \quad (29)$$

We obtained the last inequality as follows. Property (P.1) required by the PRELIM-SEEDSET tells us that if a node u_i is on at timestep later than $\theta_k - 1$ in activation sequence T_0 , then it follows that $\sum_{t < \theta_k - 1} x_{i,t} < \frac{1}{12(1 + \epsilon)}$. In this lemma we are concerned only with node u_i that are on at timestep later than $\theta_k - 1$ in activation sequence T_{k-1} . However, recall the relationship between T_0 and T_k ; any node that is on after $\theta_k - 1$ in T_{k-1} must be on after $\theta_k - 1$ in T_0 as well. Thus, we can conclude that all of the u_i 's we consider in this lemma have $\sum_{t < \theta_k - 1} x_{i,t} < \frac{1}{12(1 + \epsilon)}$. \square

Finally, we apply the arithmetic Lemma H.9 to the giant product obtained when we substitute (26) and (25) back into our original equation (23) and obtain:

$$\begin{aligned} &\min \left\{ (1 - 4(1 + \epsilon)\omega_i), \prod_{u_j \in \beta(i, \theta_k - 1)} (1 - 4(1 + \epsilon)\omega_j) \right\} \\ &\leq (1 - 4(1 + \epsilon)\omega_i)^{1/2} \prod_{u_j \in \beta(i, \theta_k - 1)} (1 - 4(1 + \epsilon)\omega_j)^{1/2} \\ &\leq \left(1 - 4(1 + \epsilon) \sum_{t < \min\{\theta_k - 1, \theta(u_i)\}} x_{i,t} \right)^{1/2} \left(\prod_{u_j \in \beta(i, \theta_k - 1)} \left(1 - 4(1 + \epsilon) \sum_{\substack{t: \\ X_{j,t} \in \beta(i, \theta_k - 1)}} f_{i, \theta_k - 1}(X_{j,t}) \right) \right)^{1/2} \\ &\leq \sqrt{1 - 2(1 + \epsilon) \left(\sum_{t < \min\{\theta_k - 1, \theta(u_i)\}} x_{i,t} + \sum_{j \in [n]} \sum_{t: X_{j,t} \in \beta(i, \theta_k - 1)} f_{i, \theta_k - 1}(X_{j,t}) \right)} \quad (\text{By Lemma H.9}) \\ &= \sqrt{1 - 2(1 + \epsilon) \sum_{t < \theta(u_i)} x_{i,t}} \quad (\text{By (28)}). \\ &\leq 1 - (1 + \epsilon) \sum_{t < \theta(u_i)} x_{i,t} \end{aligned}$$

The claim follows. \square

H.6.1 Omitted proofs from Proof of Lemma H.7

of Fact H.1. Recall that we let $I(Y)$ be an indicator function that is 1 if and only if Y is true. The border flow function $f_{i,t}(\cdot)$ can be re-written as

$$f_{i,t}(X) = \sum_{j: \text{border}(\mathcal{P}_j)=X} |\mathcal{P}_j| = \sum_{j \leq q} I(X = \text{border}(\mathcal{P}_j)) |\mathcal{P}_j|. \quad (30)$$

Notice also that we have

$$|\mathcal{P}_j| = \sum_{X \in V(\mathcal{H})} I(X = \text{border}(\mathcal{P}_j)) |\mathcal{P}_j|. \quad (31)$$

Therefore,

$$\begin{aligned} \sum_{j \leq q} |\mathcal{P}_j| &= \sum_{j \leq q} \left(\sum_{X \in V(\mathcal{H})} I(X = \text{border}(\mathcal{P}_j)) |\mathcal{P}_j| \right) \quad (\text{By Equation 31}) \\ &= \sum_{X \in V} \sum_{j \leq q} I(X = \text{border}(\mathcal{P}_j)) |\mathcal{P}_j| \\ &= \sum_{X \in V} f_{i,t}(X) \quad (\text{By Equation 30}) \\ &= \sum_{X \in \beta(i,t)} f_{i,t}(X) \quad (\text{Only border nodes have none-zero value on } f_{i,t}(\cdot)). \end{aligned}$$

Finally, our claim follows from the fact that

$$\sum_{\theta(u_i) \leq t' \leq t} x_{i,t'} = |\mathcal{F}_{i,t}| = \sum_{j \leq q} |\mathcal{P}_j|,$$

□

of Another Arithmetic Lemma H.9. We have

$$\begin{aligned} &\prod_{i \leq n} (1 - x_i) \\ &= 1 - \sum_{i \leq n} x_i + \sum_{i_1 \neq i_2 \in [n]} x_{i_1} x_{i_2} - \sum_{i_1 \neq i_2 \neq i_3 \in [n]} x_{i_1} x_{i_2} x_{i_3} + \dots + (-1)^k \sum_{i_1 \neq i_2 \neq \dots \neq i_k} x_{i_1} \dots x_{i_k} + \dots \\ &\leq 1 - \sum_{i \leq n} x_i + \sum_{i_1 \neq i_2 \in [n]} x_{i_1} x_{i_2} + \sum_{i_1 \neq i_2 \neq i_3 \in [n]} x_{i_1} x_{i_2} x_{i_3} + \dots + \sum_{i_1 \neq i_2 \neq \dots \neq i_k} x_{i_1} \dots x_{i_k} + \dots \\ &\leq 1 - \sum_{i \leq n} x_i + \sum_{i_1, i_2 \in [n]} x_{i_1} x_{i_2} + \sum_{i_1, i_2, i_3 \in [n]} x_{i_1} x_{i_2} x_{i_3} + \dots + \sum_{i_1, i_2, \dots, i_k} x_{i_1} \dots x_{i_k} + \dots \\ &= 1 - \sum_{i \leq n} x_i + \left(\sum_{i \leq n} x_i \right)^2 + \left(\sum_{i \leq n} x_i \right)^3 + \dots \\ &= 1 - \sum_{i \leq n} x_i + \frac{(\sum_{i \leq n} x_i)^2}{1 - \sum_{i \leq n} x_i} \\ &= 1 - \sum_{i \leq n} x_i \left(1 - \frac{\sum_{i \leq n} x_i}{1 - \sum_{i \leq n} x_i} \right). \end{aligned}$$

□

H.6.2 Obtaining Lemma 3.9 from Lemma H.7

of Lemma 3.9. Let us write $I(X)$ be an indicator variable that sets to 1 if X is true. We have

$$\hat{\rho} = \sum_{u_i: T_{k-1}(u_i) \geq \theta_k} I(T_k(u_i) < \theta_k).$$

We start by computing $E[\hat{\rho}]$:

$$\begin{aligned} E[\hat{\rho}] &= \sum_{u_i: T_{k-1}(u_i) \geq \theta_k} E[I(T_k(u_i) < \theta_k)] \\ &\geq (1 + \epsilon) \sum_{u_i: T_{k-1}(u_i) \geq \theta_k} \sum_{t < \theta_k} x_{i,t} \quad (\text{Using Lemma H.7}) \\ &= (1 + \epsilon)\gamma \quad (\text{Definition of } \gamma). \end{aligned}$$

To prove the inequality we take:

$$\begin{aligned} (1 + \epsilon)\gamma &\leq E[\hat{\rho}] \\ &\leq \Pr[\hat{\rho} \geq \gamma] \cdot n + \gamma. \end{aligned}$$

and algebra shows that $\Pr[\hat{\rho} \geq \gamma] \geq \epsilon \cdot \gamma/n$. □